12-13-2003

# Remote Control of Heating, Ventilating, and Air Conditioning System with Labview

Yu-Loong Liew

# REMOTE CONTROL OF HEATING, VENTILATING, AND AIR CONDITIONING SYSTEM WITH LABVIEW

By

Yu-Loong Liew

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Electrical Engineering
in the Department of Electrical and Computer Engineering

Mississippi State University

December 2003

REMOTE CONTROL OF HEATING, VENTILATING, AND AIR CONDITIONING

SYSTEM WITH LABVIEW

By

Yu-Loong Liew

Approved:

Randolph Follett
Assistant Professor of Electrical
and Computer Engineering
(Major Professor)

Roger King
William L. Giles Distinguished
Professor of Electrical and
Computer Engineering
(Committee Member)

Nicholas Younan
Professor of Electrical and
Computer Engineering
(Graduate Program Director and
Committee Member)

A. Wayne Bennett
Dean of the Bagley College of
Engineering

Name: Yu-Loong Liew

Date of Degree: December 13, 2003

Institution: Mississippi State University

Major Field: Electrical Engineering

Major Professor: Dr. Randolph Follett

Title of Study: REMOTE CONTROL OF HEATING, VENTILATING, AND AIR CONDITIONING SYSTEM WITH LABVIEW

Pages of Study: 88

Candidate for Degree of Master of Science

Recent technological advances have extended the use of some old technology such as the X-10 home automation. With the use of X-10 and the Internet, a remote control home is possible. One of the major energy consumption appliances in a house is the heating, ventilating, and air conditioning system. The main objective of this study is to explore the possibility of using X-10 and the Internet to serve as an energy saving system. A simulation is used to estimate the energy consumption.

# DEDICATIONS

I would like to dedicate this research to my family and my fiancée Judy Liaw. Thank you for all your support.

# ACKNOWLEDGMENTS

I would like to express my sincere gratitude to many people whom have helped me with this thesis. First of all, I would like to thank Dr. Randolph Follett, my major professor, for spending his time and effort to guide me through this thesis process. Appreciations are also due to Dr. Roger King and Dr. Nicholas Younan for their aid and directions.

# TABLE OF CONTENTS

CHAPTER Page

APPENDIX Page

# LIST OF TABLES

TABLE Page

# LIST OF FIGURES

FIGURE Page

| FIGURE | Page |
| --- | --- |

# CHAPTER I

# INTRODUCTION

The concept of "home automation" or "smart home" is not new. In fact, integrated home systems have been offered since the 1960s [1]. There are many regional and international standards available [1,2]. These standards include X-10, Smart House, LonWorks, Consumer Electronics Bus (CEBus), Home Bus System (HBS), TRON, BatiBUS, EHS, EIB, and HES.

All of these standards are not compatible with one another. Therefore, most people do not incorporate these technologies to their home. Furthermore, most of the industries invest in developing communication networks, but not much on the applications or appliances themselves [1].

This thesis will first compare and contrast some home automation standards that are available in the United States. Next, the possibility of remote control using LabView and Internet to control such devices is proposed. The X-10 devices used for this purpose will be mainly for heating, ventilating, and air conditioning (HVAC) system. Then, a model is developed and simulated to investigate the monthly cost of utilizing such system. Finally, the conclusion will discuss how the X-10 system performed in comparison to a regular HVAC system. Possible future work will be discussed at the end of the thesis.

# CHAPTER II

# HOME AUTOMATION STANDARDS

## 2.1 Overview

There are many regional and international standards to control home appliances. In this thesis, only LonWorks, CEBus, and X-10 standards will be examined. As mentioned before, none of these standards are compatible with each other.

## 2.2 LonWorks Standard

LonWorks was developed by Echelon. It is a complete solution to automation for the building automation industry. Therefore, it is a very *powerful* home automation standard. It is capable of controlling up to 32,000 devices, "and can be used in everything from supermarkets to petroleum plants, from aircraft to railway cars, from fusion lasers to slot machines, from single family homes to skyscrapers" [6]. LonWorks systems are based on the concepts [4]:

- All control systems are the same regardless of the applications,
- A networked control system is more powerful, flexible, and scaleable than a non-networked system, and
- Businesses can save more money with networked control systems in the long term than with a non-networked system.

Each LonWorks device has a neuron chip, and the chip is made out of three 8-bit inline processors. Two of the processors are optimized for executing the communication protocol, while the other is for the node's application. Therefore, the neuron chip is both network processor and application processor.

The LonWorks communication protocol is the heart of the LonWorks system. It is called the LonTalk protocol. This protocol allows devices in the system to send and receive commands without any knowledge of the topology of the system. It is a layered, packet-based, peer-to-peer communication protocol like the Ethernet and Internet protocol. Both Ethernet and Internet are designed to send and receive vast amounts of data over the networks; however, LonTalk is designed to send or receive control signals for a control system.

Since the LonWorks system is a very powerful system that is capable of controlling a building or a product line of a factory, the price to install the system is very expensive. For the industry, it might not be expensive to them; however, for homeowners to implement this system, it will be too expensive. For example, the i.LON™ 1000 Internet Server Starter Kit will cost about $3000. This kit includes an internet server, three LonWorks modules, a PCI card for a desktop or a PC card for laptop, software, and other accessories. With this startup kit, all the hardware for setting up a simple home automation system is included. The LonMaker Integration Tool included uses Microsoft Visio as a user interface, therefore controlling the system will require simple block diagrams drawings. The i.LON Internet server will be the main component for connecting the LonWorks system from the home network to the Internet. The Internet server has all the necessary tools for setting up an Internet server.

## 2.4 CEBus Standard

The Electronic Industry Association (EIA) developed the Consumer Electronic Bus (CEBus) standard. The CEBus network architecture uses the open systems interconnection (OSI) layered network architecture. The OSI network has seven layers, which includes Physical, Data Link, Network, Transport, Session, Presentation, and Application. However, the CEBus network only utilizes the Physical, Data Link, Network, and Application layers.

There are six different mediums of transmissions for the CEBus physical layer that consists of the power line (PL), twisted pair (TP), infrared (IR), radio frequency (RF), coaxial (COAX), and fiber optics (FO). Up until now, EIA has only released four specifications for the CEBus transmission medium. The COAX and the RF transmission mediums are still in the research phase [5]. In this thesis, only the PL communication medium is examined. Brief information about the other mediums of transmission can be found in [5].

The CEBus power line specification is based on Amplitude Shift Keying (ASK) with bursts of 120 kHz signals on the power line. The state when a 120 kHz burst is present is called "superior state", while the absence of the burst is called "inferior state." The PL uses a unit symbol time (UST) of 1ms. A logic "1" is 1 UST, a "0" is 2 UST, an end of frame (EOF) is 3ms long, and an end of packet (EOP) is 4ms long. A NULL symbol of 125ms is inserted at every 158ms of continuous activity. The NULL symbol is important to ensure that other PL standards will not interfere with CEBus devices. Other PL standards send their control signals in a continuous fashion; therefore, the CEBus will just ignore other PL standards' control signals. The downside to the NULL symbol is that it wastes the channel bandwidth and lowers the maximum throughput from 1000b/s to 560b/s [5].

Due to the slow data transfer in the PL specification, a spread spectrum-signaling standard is added to the PL specification. Two "superior" states are defined as SUPERIOR01 and SUPERIOR02. The control signals corresponding to these states are 180 degrees out of phase. This new method of transmission increased the maximum data rate from 1kb/s to 10kb/s [5].

Though the CEBus standard seems to be a good home automation standard, currently there are no products available on the market. This is probably due to the fact that not all of the physical mediums are defined, and the CEBus technology is relatively new compared to other power line communication standards. However, once this standard is well developed, it might be more suitable to use as a home automation standard than Echelon's LonWorks since this standard is designed specifically for home automation.

## 2.5 X-10 Standard

The X-10 standard is developed and patented by X-10 Inc. X-10 is the most popular choice among the three home automation standards because there are a lot of products that a home automation enthusiast can purchase. Even though X-10 and one of the CEBus physical specifications run on a power line, they are not compatible with one another. The X-10 standard only uses the power line as transmission medium. On some of the X-10 product lines, users can control X-10 devices using a radio frequency remote control supplied by X-10.

X-10 transmissions are synchronized to within 200µs of the zero crossing point of the AC power line [6]. The X-10 transmitters provide a 60Hz square wave with a maximum delay of 100µs from the zero crossing of the 120V voltage line.

A logic 1 is represented by a 1ms burst of a 120 kHz signal and a logic 0 is represented by the absence of 120 kHz signal. At each zero crossing or every 8.333ms, three bits are transmitted. Figure 2.1 shows how the signal looks. These three bits are used to coincide with all three phases in a three-phase distribution system.

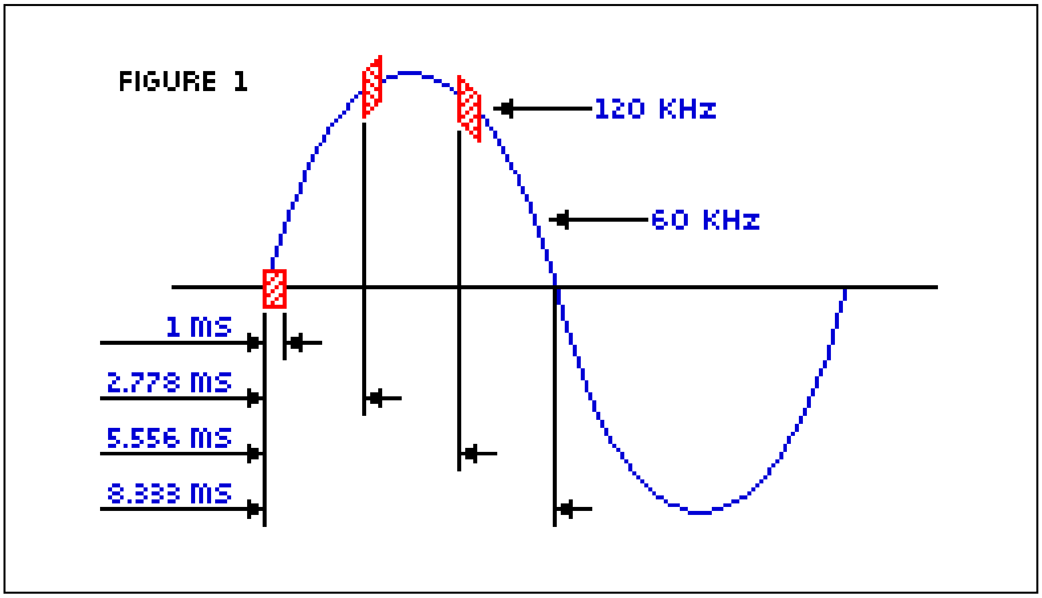



Figure 2.1 Transmission signal of X-10[1].

A complete X-10 transmission encompasses 11 power line cycles, with a repeat after the first transmission. The transmission begins with two cycles of start code (1110) followed by four cycles of house code, and five cycles of either number code or function code. Figure 2.2 shows the code transmission when either a device code or a function is sent. Within each block of data, the data bits are transmitted in true and complement form on alternative half cycles. For example, if a 1ms burst of signal is available for one half of the cycle then there will be no signal for the second half of the cycle.

[1] Diagram obtained from www.x-10.com

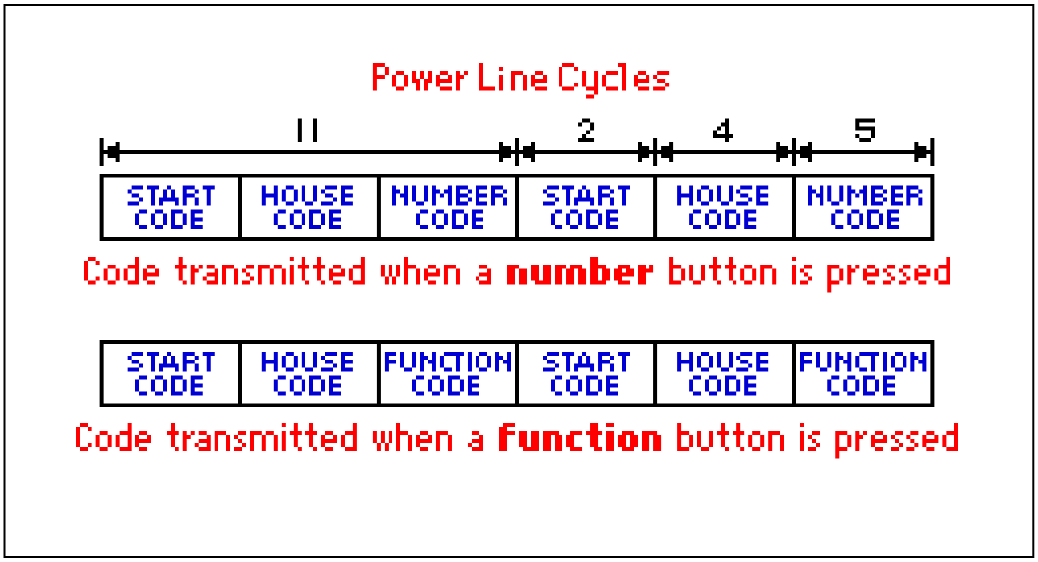



Figure 2.2 Transmitted code for selecting a function or a number[2].

There are 16 different house codes, number codes and function codes. Hence, we can control up to 256 X-10 devices with all the house codes and number codes used. For the time being, X-10 standard is the "de facto" standard for power line carrier transmission because of both availability and cost. An X-10 starter kit with an appliances module, a lamp module, a PC to X-10 interface, software, and two remote controls cost about $50. The X-10 bi-directional heating, ventilating and air conditioning (HVAC) system costs about $250. The system consists of a wall display unit and an X-10 HVAC control unit. The wall display unit provides the regular thermostat user interface for current temperature display and temperature set point changes. The X-10 control unit controls the standard heating and cooling system via the thermostat connection.

[2] Diagram obtained from www.x-10.com

## 2.6 Conclusion

The X-10 standard is chosen because its products are easily available at an affordable price. Once all the CEBus standards are well defined and its products are readily available, CEBus might be a better choice. CEBus has more communication mediums than X-10 and it is specifically designed to be used in a home unlike LonWorks which is an industrial standard that turned into home use.

One important factor that one has to consider before installing any home automation system is the price of the installation and its products. Though LonWorks could be the most robust system for home automation, the installation for the system is too expensive. One not only has to install the system, but also has to install wiring for the LonWorks network.

The power line communication network seems to be a good home automation network on the surface but there are some drawbacks to the network [5]. A PL network consists of all homes wired to the same distribution transformer. Signals from one house can propagate to another house causing interference. A filtering network maybe used to filter out the interference, but the filter might also filter out signals used to control modules in the house. It will make the X-10 communication system less reliable. Furthermore, most houses are wired in a two-phase or three-phase power system. Devices connected on phase 1 cannot communicate with devices on the other phases. The communication can only be accomplished if a signal coupler is present. So far, only CEBus has specified the construction of such devices. For X-10 devices, the user has to check the phase of the wiring by trial and error.

# CHAPTER III

# CONTROLLING X-10 DEVICES USING LABVIEW AND THE INTERNET

## 3.1 Overview

In this thesis, National Instruments' LabView software is used to control X-10 devices. LabView is a graphical programming language development software. The software uses graphical icons instead of lines of text to create applications. This can reduce the time to write and debug programs.

Most X-10 devices come with software to control its devices. However, the provided software is not capable of controlling X-10 devices from the Internet. Typically, additional software is needed for such function. This makes remote control of the X-10 devices difficult. Hence, the use of LabView is proposed to control X-10 devices.

## 3.2 Software selection

LabView is chosen instead of other programming languages because of its ease of programming and debugging. LabView is a graphical programming language. Almost all other programming languages use lines of text of codes to create applications. This can make creating an application and debugging it difficult and time consuming.

A typical LabView program consists of a front panel and a block diagram. Each program is called a virtual instrument (VI), and a sub program is called a sub-VI. Figure 3.1 and Figure 3.2 shows a front panel of a serial communication VI, and the corresponding block diagram of the VI. For more information about how to program VIs, refer to [7].

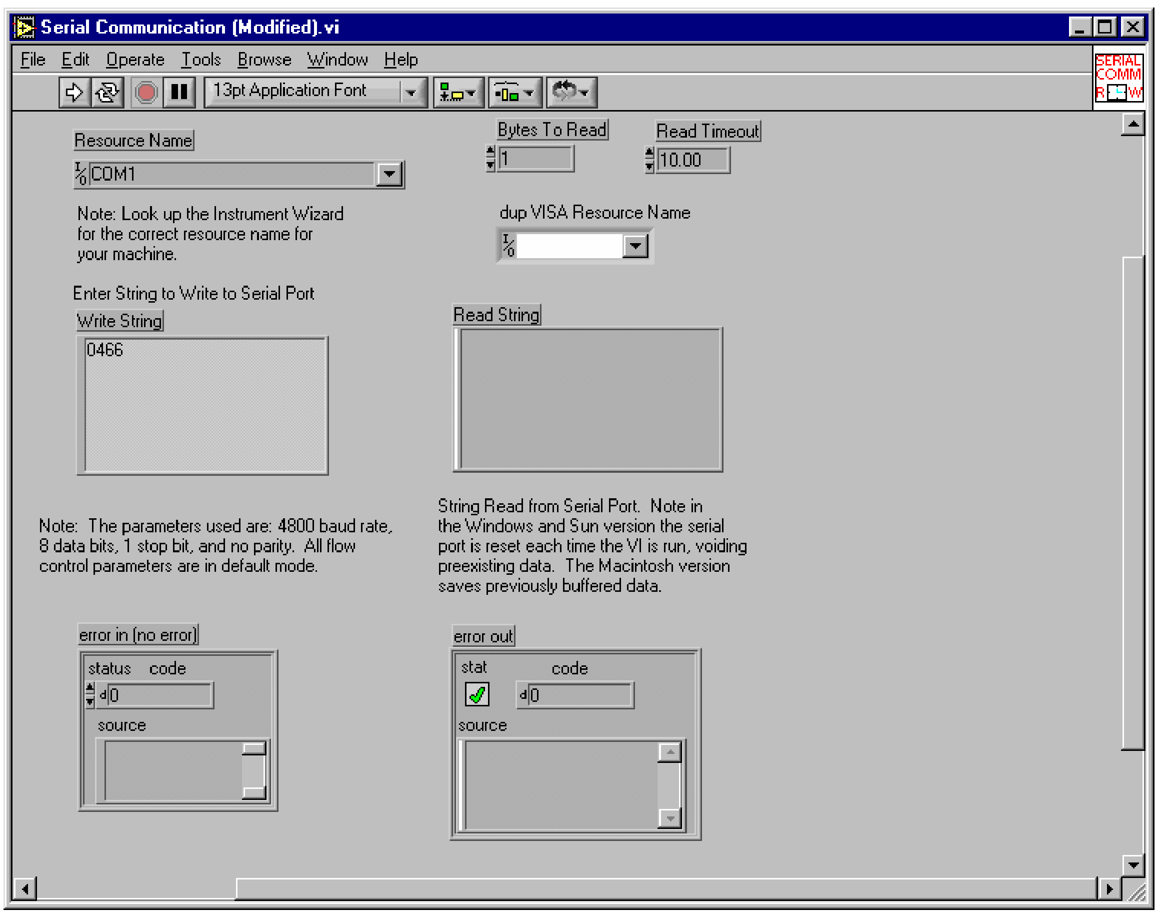



Figure 3.1 Front Panel of the Serial Communication (Modified) Virtual Instrument.

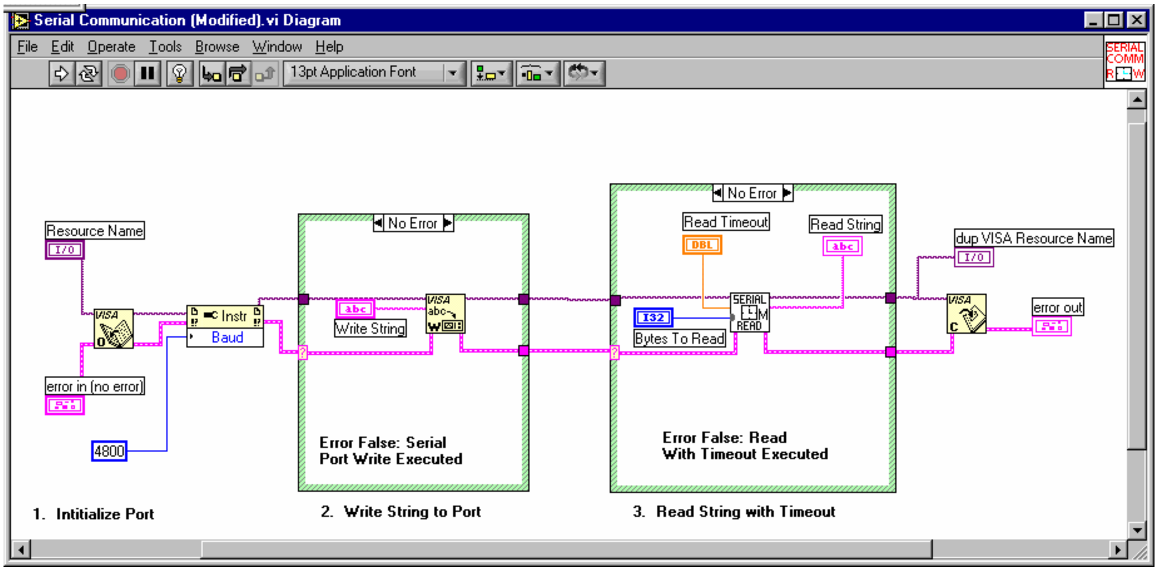



Figure 3.2 Block Diagram of the Serial Communication (Modified) Virtual Instrument.

LabView also has add-on software that is capable of implementing a web server. The web server is based on the common gateway interface (CGI). CGI was the first technology that allows users to call external routines remotely regardless of the platform. With CGI, a user can control VI's remotely using a web browser. CGI is not the only Web-based control that is available for LabView. Java and ActiveX technologies can also be used to control VI's through the Internet. However, with the latter two technologies, third party software is needed. Therefore the CGI technology is used.

X-10 devices can be easily controlled through the Internet using LabView and the Internet Toolkit. One of the drawbacks of using LabView is the need to write programs. The software provided together with X-10 devices does not require any complex programming. The only programming needed for such software is to setup macros to perform certain tasks; for example, to turn on a set of lights at night. Nevertheless, by writing programs using LabView, there is more flexibility in how the device is controlled, what user interface is used, and so on.

Another major disadvantage of using LabView is the price of the software itself. Due to its ease of programming and debugging, LabView is significantly more expensive than other programming languages. For example, a copy of Microsoft Visual Studio .NET Professional Edition is only about $1100 and C++ Builder 6 Professional – New User version is only $1000. For a base package, LabView costs about $1000. For the full development package, it costs about $2000, and the LabView professional version costs about $3500. These prices do not include the Internet Toolkit which costs about $500. The prices listed above are the initial purchase price. The upgrade price is about $400 for version upgrade. Note that this upgrade only applies to base package to base package upgrade, full development to full development upgrade, and so on. For a home user, the LabView full development package will be needed since controlling X-10 devices requires the use of RS-232 port, and only the full development version and above have such drivers available. Therefore the cost of LabView programming software and an X-10 system with HVAC module comes out to be about $2750. It turns out to be slightly cheaper than the LonWorks system.

### 3.3 Web security

Web security is a primary concern when dealing with the Internet. Therefore, we have to discuss web securities available with the LabView server. This section will not discuss all the available tools or ways to prevent hackers or crackers to attack our network, but it provides an introduction to this complex topic.

Since we plan to have a server that is connected to the Internet or through a dial-in line, we need to look at remote access security. According to [8], remote access security revolves around the authentication, authorization, and accounting (AAA) model.

**Authentication** looks at who the user or entity is. Common authentication implementation is with a username/password scheme.

**Authorization** determines what each user can do on the remote machine. This is very important to ensure that not everybody can delete files from the machine.

**Accounting** records what the user has been doing. This allows the administrator to look at how users are accessing the system.

There are a few ways to implement the AAA model in LabView. We can write special VI to control the authentication to the system by checking the IP address or we can use the Internet Toolkit Web server to set up and manage users, groups, and passwords. For more information about how to configure and write special VIs, one can consult [8].

### 3.4 Sample application with X-10 devices

3.4.1 Overview

A simple application is used to show the capabilities of X-10 devices and LabView web server. The application is to maintain the temperature of an aluminum block at about 45ºC. The temperature of the aluminum block is obtained using a thermistor. The block is heated using a solder iron connected to an X-10 device. CM11A is used as a bridge between the PC and X-10 devices.

The set point of the aluminum block can be changed from the Internet, however, the program can only handle a small change in temperature. For example, the program will work well with one or two degrees changed from the default value of 40ºC. When the set point is farther from the default value, the program will not be able to maintain the aluminum block close to the set point. This is mainly due to the program not tuned for a wide range of temperature control. A wide range control scheme will require a more complex control scheme than the one being used for this simulation.

3.4.2 Thermistor

A thermistor is used to measure the temperature of the aluminum block because it is more sensitive than thermocouples and resistance temperature devices (RTD) [9]. A thermistor is a temperature sensitive resistor. The thermistor resistance-temperature relationship is negative; the resistance decreases with increasing temperature. A typical resistance-temperature relationship of a thermistor is shown in Figure 3.3. Thermistor costs less than a dollar, thermocouples cost about \$20 for a spool of 50 feet of thermocouple wire, and an RTD costs about \$20.

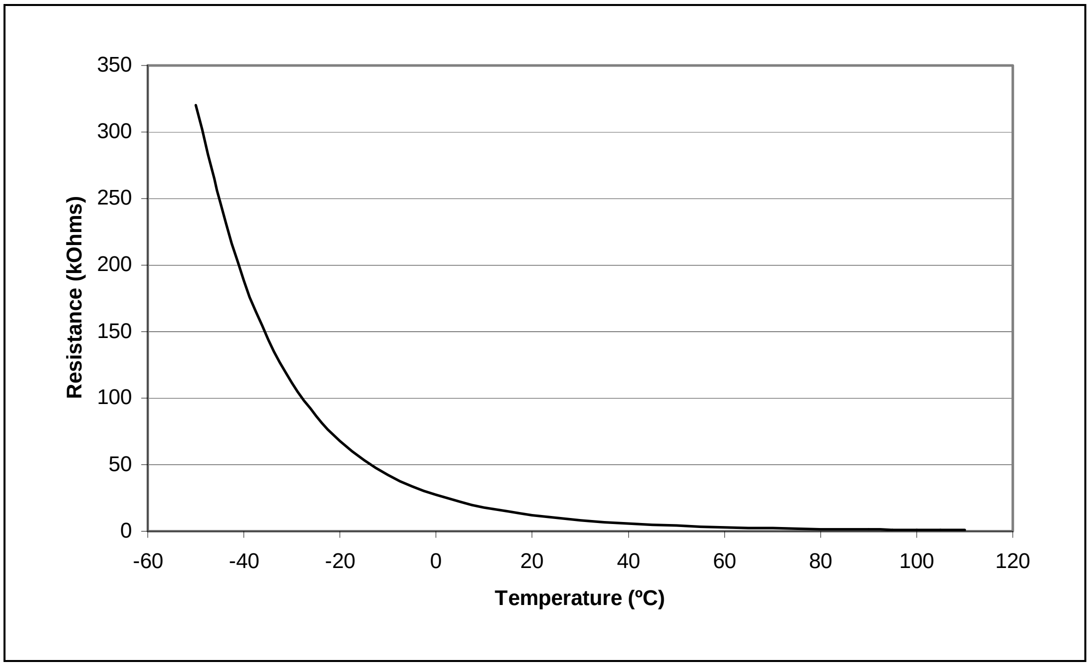



Figure 3.3 Typical resistance-temperature relationship of a thermistor.

A thermistor is an extremely non-linear device. Therefore, manufacturers have not standardized thermistor curves like RTD or thermocouple curves. However, a thermistor curve can be approximated using the Steinhart-Hart equation:

$$\frac{1}{T} = A + B\ln(R) + C(\ln(R))^3 \tag{3-1}$$

where:

| | | |
|---|---|---|
| T | = | temperature [ Kelvin], |
| R | = | resistance of the thermistor [kΩ], |
| A, B, C | = | curve-fitting constants. |

A, B, and C are found by selecting three data points from the manufacturer's published data curve, and solving the three simultaneous equations. If the data points are chosen to span less than 100ºC within the nominal center of the thermistor, the accuracy of the equation approaches ±0.02ºC. According to [9], the operating temperature for a thermistor is from -40ºC to 150ºC. For Equation (3-1) to have an accuracy of ±0.02ºC, the temperature selected has to be within the operating temperature and the selected temperature, T1, T2, and T3, has to be within 50ºC with one another. In other words, $-40^{\circ}\mathrm{C} \leq \mathrm{T1, T2, T3} \leq 150^{\circ}\mathrm{C}$, $|T1 - T2| \leq 50^{\circ}\mathrm{C}$, $|T2 - T3| \leq 50^{\circ}\mathrm{C}$, and $|T1 - T3| \leq 100°C$. The nominal center of the thermistor is therefore about 50ºC from the two upper and lower temperature selected for the curve constant computation. The accuracy of Equation (3-1) is too ideal. Such accuracy can only be attained using exact values for temperatures and resistances together with an accurate algorithm to compute the curve constant. An error of about ±0.5ºC is more reasonable.

The curve constants used for the LabView VI are computed using Equation (3-1). Three data points selected from the manufacturer's data curve are 20ºC with resistance of 1.86kΩ, 30ºC with resistance of 1.374kΩ, and 46ºC with resistance of 0.794kΩ. From these data points, the three curve-fitting constants obtained are A = 0.0036933569, B = -0.0002549387, and C = 0.0000038370. Figure 3.4 shows the actual data points of the thermistor against the calculated data points using the curve fitting constants above.

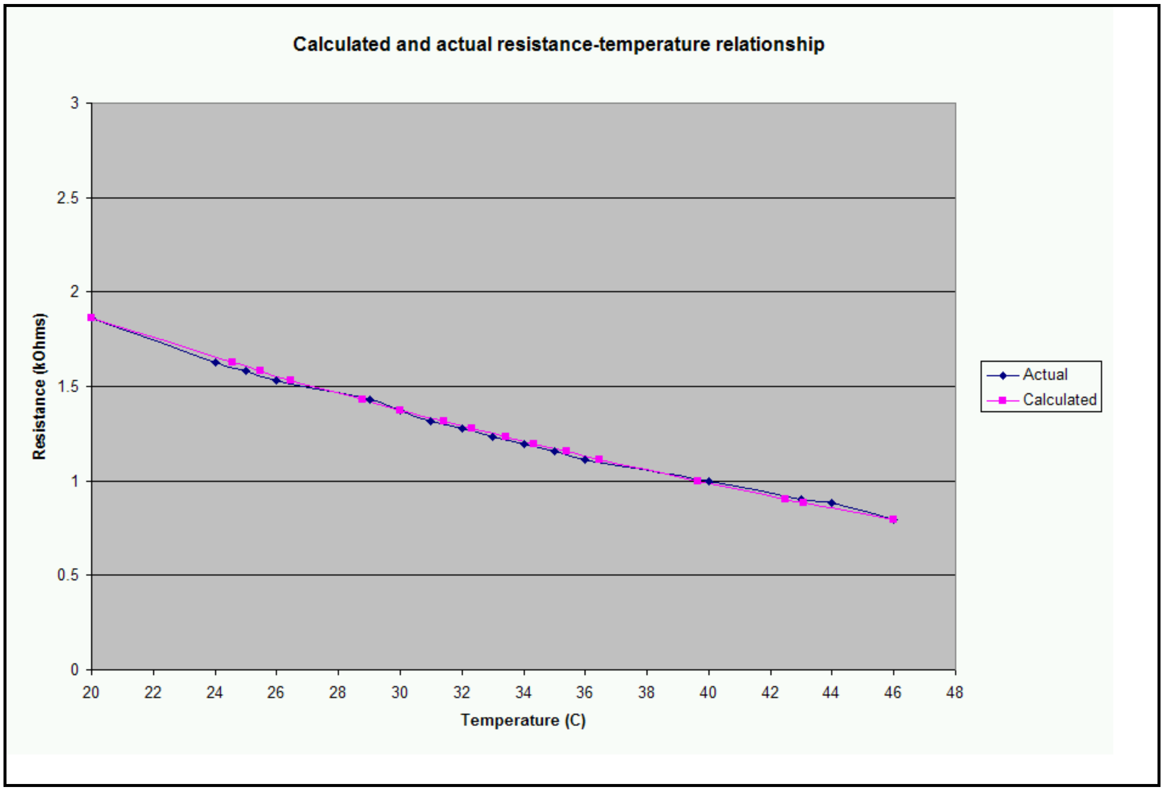



Figure 3.4 Comparison of the actual and calculated data points for the thermistor.

To measure temperature, the thermistor is connected to a power supply together with a current limiting resistance. This circuit is connected to a personal computer data acquisition card to measure its voltage. LabView is used to convert the measured voltage to a temperature reading using Eq (3-1). The end result has an error of about ±0.5ºC, however the purpose for the thermistor is to imitate a thermostat. Therefore, the accuracy of the temperature reading is not of great importance. For accurate temperature reading, thermocouples or RTD's should be used. However, both thermocouples and RTD's are less sensitive than thermistor. The circuit diagram to obtain the temperature reading is shown in Figure 3.5.

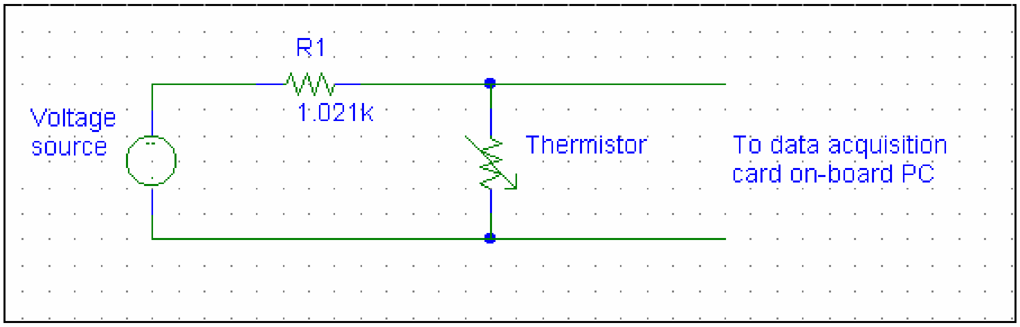



Figure 3.5 Circuit used to measure the temperature from a thermistor.

### 3.4.3 X-10 CM11A

The X-10 CM11A communication device is an important communication bridge between a PC and the X-10 devices. This device is capable of sending and receiving X-10 commands through power line. The house codes from A to P, device codes from 1 to 16, and function codes with respect to binary values are listed below:

Table 3.1 Binary values for house codes, device codes, and function codes.

| ***House code*** | ***Device code*** | ***Function*** | ***Binary value*** |
|---|---|---|---|
| A | 1 | All Lights Off | 0110 |
| B | 2 | Status Off | 1110 |
| C | 3 | On | 0010 |
| D | 4 | Pre-set Dim (1) | 1010 |
| E | 5 | All Units On | 0001 |
| F | 6 | Hail Acknowledge | 1001 |
| G | 7 | Bright | 0101 |
| H | 8 | Status On | 1101 |
| I | 9 | Extended Code | 0111 |
| J | 10 | Status Request | 1111 |
| K | 11 | Off | 0011 |
| L | 12 | Pre-set Dim (2) | 1011 |
| M | 13 | All units off | 0000 |
| N | 14 | Hail Request | 1000 |
| O | 15 | Dim | 0100 |
| P | 16 | Extended Data Transfer | 1100 |

The CM11A is connected to the PC through the serial port available on the PC. The serial parameters for the communication between the CM11A interface and PC are shown below:

| | |
|---|---|
| Baud rate: | 4800bps |
| Parity: | None |
| Data Bits: | 8 |
| Stop Bits: | 1 |

An X-10 transmission from the PC to the interface refers to the communication of a combination of house code and device code, or the transmission of a function code. The format for these transmissions is listed in Table 3.2. This format applies to all transmission between the PC and the interface. The only difference is the first transmission from the PC.

Table 3.2 X-10 transmission format from PC to CM11A interface

| ***Number of bytes*** | ***PC*** | ***Interface*** |
|---|---|---|
| 2 bytes | Header:Code | |
| 1 byte | | Checksum |
| 1 byte | Acknowledge | |
| 1 byte | | Interface ready to receive |

The Header byte defined as:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Header: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | <Number of Dims> | | | | | 1 | F/A | E/S |

Number of Dims defines the dimming function of an X-10 module. This function is only available for a lamp module. Number of Dims is a value between 0 and 22 identifying the percentage of dims to be transmitted with 22 being 100%. Bit 2 is always set to ‘1’ to make sure that the interface is able to maintain synchronization. F/A defines whether the following byte is a function (1) or an address (0). E/S defines if the following byte is an extended transmission (1) or a standard transmission (0). The Code byte is defined as:

| Code: | 7 6 5 4 | 3 2 1 0 |
|---|---|---|
| Address: | < House code > | < Device code > |
| Function: | < House code > | < Function > |

Note that function only operates with devices addressed with the same house code. Once the interface receives a transmission from the PC, it will sum all of the bytes, and return a byte called a checksum. If the checksum is correct, the PC should acknowledge by returning a value of hex 0 (0x00) to indicate that transmission should take place. However, if the checksum is incorrect, the PC should attempt to re-transmit the Header:Code combination, and wait for a new checksum. Once the X-10 transmission has taken place, the interface will send 0x55 to the PC to indicate that the interface is ready for another transmission. Below is an example communication between the PC and the CM11A interface. The example below will turn on an X-10 module with address of house code A and device code 1, or in short address A1.

| *PC* | *Interface* | *Description* |
|---|---|---|
| 0x04, 0x66 | | Address: A1 |
| | 0x6A | Checksum ((0x04 + 0x66) & 0xFF) |
| 0x00 | | OK for transmission |
| | 0x55 | Interface ready |
| 0x06, 0x62 | | Function: A ON |
| | 0xEE | Incorrect checksum |
| 0x06, 0x62 | | Re-transmit function |
| | 0x68 | Checksum ((0x06 + 0x62) & 0xFF) |
| 0x00 | | OK for transmission |
| | 0x55 | Interface ready |

The first transmission, 0x04, is the header byte which is represented by binary 0100. By referring to the definition of header byte above, 0x04 signals the CM11A interface that the following command is an address command with standard transmission method. The second transmission byte is the address A1. The interface will reply with a checksum as defined above. Next, the software communicating with the PC interface needs to send a hex 0 to confirm that the transmission is correct, and the interface will reply with hex 55 indicating that the interface is ready for additional transmission. Hex 6 (0x06) lets the interface to know that a function will be send next. The function sent to the CM11A interface is a command of all the devices with house code A to be turned on. Note that the checksum is wrong initially, and the software has to re-transmit the function to the interface.

More detailed information on how to send extended X-10 signals from the PC to the interface, and on how to obtain data from the interface can be found in [10].

### 3.4.4 Sample application

A sample application is used to test the LabView's web server and the CM11A X-10 module. The application allows the user to change the temperature set point for an aluminum block connected to a soldering gun. This application can be used to represent the control of a heating system.

The control scheme used in this application is shown in Fig 3.6. The initial decision box is used to make sure that there is enough temperature range for the initial heating process. If this not implemented, then the soldering iron will always turn on whenever the software is launched. This will make it almost impossible to control the temperature of the soldering iron. If the difference between the temperature set point, Tsp, and the initial temperature, Ti, is less than 18°C, then the “temperature difference”, Ta, is set to 13°C. However, if the temperature difference between the initial temperature and the set point temperature is more than 18°C, then Ta is set to (Tsp – (Ti + 5)). Hence the soldering iron will be ON when the current temperature, Tc, is less than or equal to (Tsp - 13°C) or (Tsp – (Ti + 5°C)) depending on the Ti.

Once the initial heating is done, the software moves to the second phase of the control scheme. In this phase, whenever the temperature of the soldering iron is between (Tsp – 1.42) and (Tsp – 1.6), the soldering iron will be turned ON. The second phase will make sure that the temperature of the soldering iron is maintained between Tsp ± 2°C.

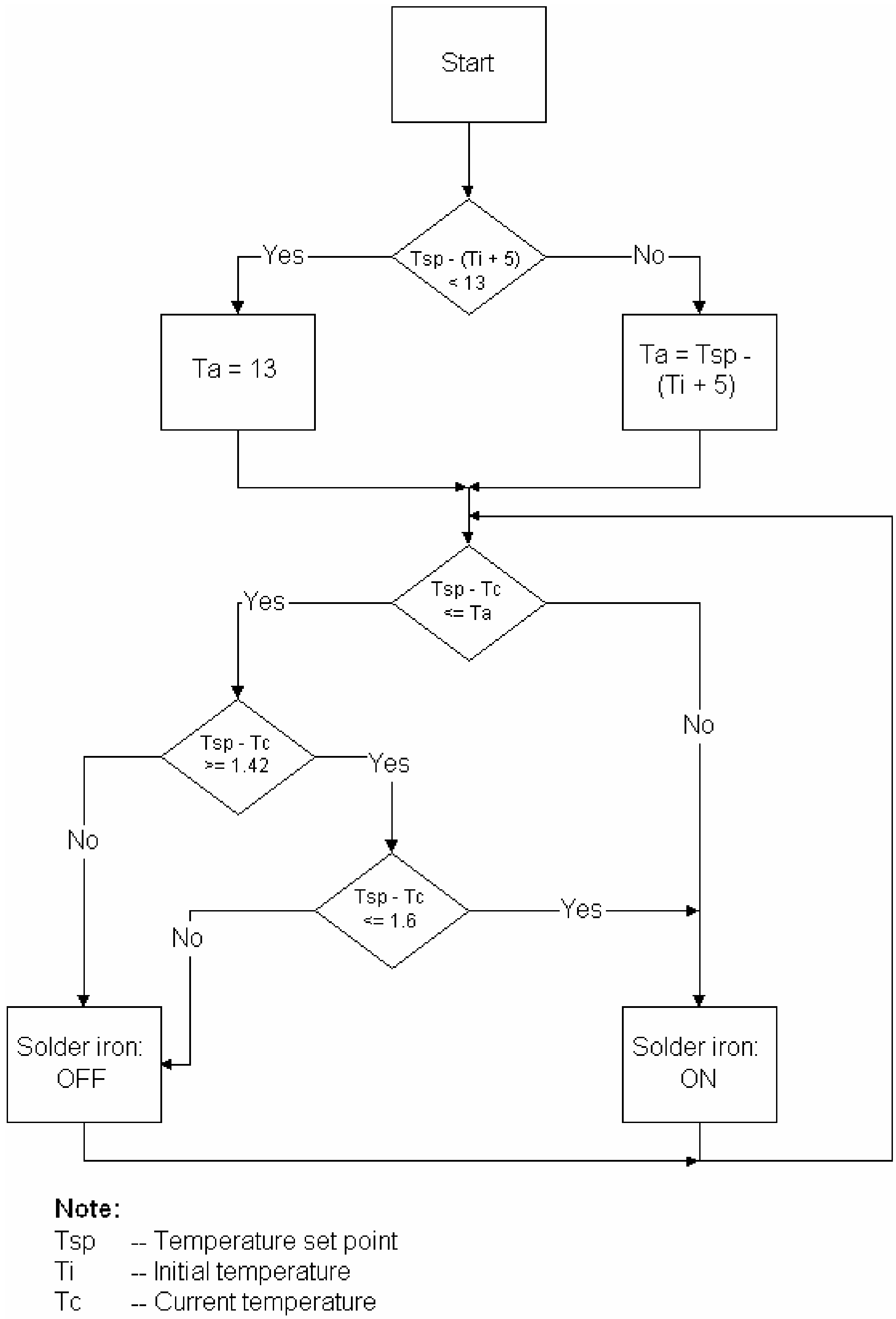



Figure 3.6 Flowchart of the control scheme for the sample application.

The front panel of the VI is shown in Figure 3.7. This will be the same interface seen when using a web browser. From the web browser, the user will be able to change the set point of the soldering iron from 38°C to 42°C. All the front panels together with their VI diagrams (source code) used for this sample application are shown in Appendix A.

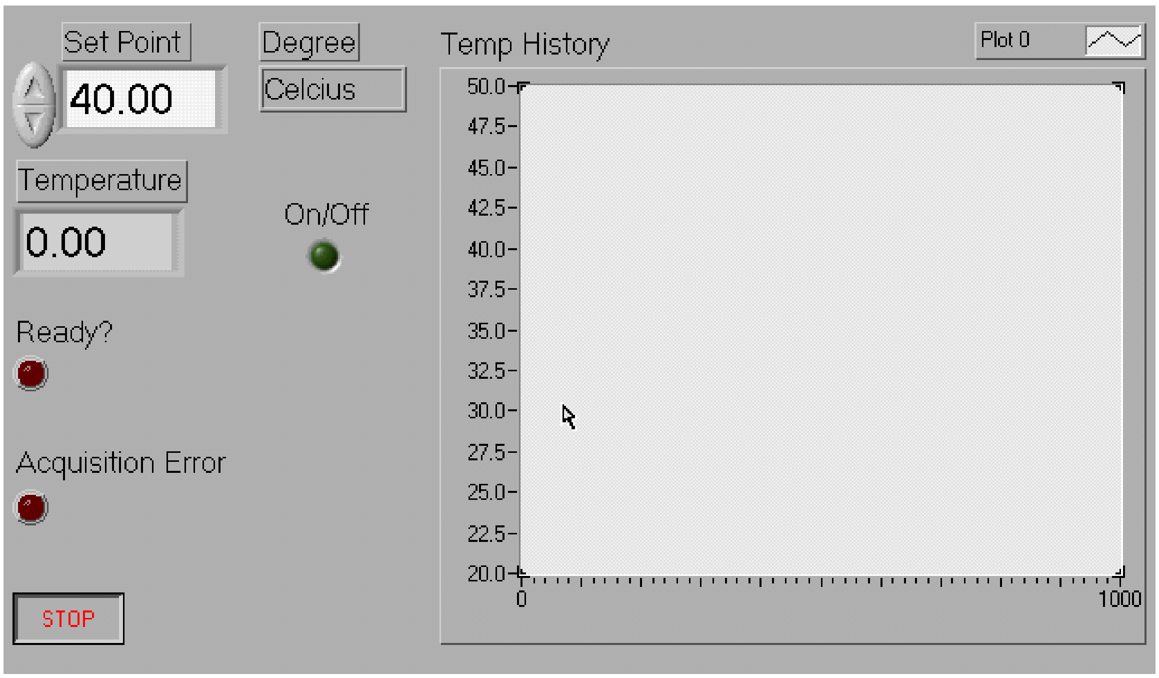



Figure 3.7 Front panel of *controlv1.vi*.

### 3.5 X-10 Thermostat

The X-10 thermostats available for remote control are the TX15 model and the TX15-B model thermostat. The TX15 model is the standard model that can receive X-10 commands, while the TX15-B model is capable of receiving and sending X-10 commands.

The TX15 consists of two units, a wall display unit and a control unit. The wall display unit acts as the traditional thermostat allowing users to monitor the current temperature and changing the set point. The control unit is connected to the heating, ventilating, and air conditioning (HVAC) system and receives X-10 commands to control the set points and thermostat modes. Settings like set point and fan mode are sent to the control unit via the wall unit or X-10 commands.

The control unit will turn on the heating system if the temperature falls one degree below the set point, and turn off when the temperature reaches the set point. The unit will turn on the cooling system if the temperature rises one degree above the set point, and turn off when the temperature reaches the set point temperature. However the control unit has a minimum run time of four minutes for both heating and cooling even though the set point temperature is reached to prevent any short heating and cooling that might damage the heating and cooling system. The control unit also has a short cycle protection delay of five minutes after any compression operations. This protection allows the compressor head pressure to bleed off before another compression operation.

The control unit is connected to the power line via an external PSC05 X-10 Interface Module. All X-10 commands sent to the control unit are decoded and displayed on the wall display unit for three seconds after which the current temperature will be shown. The thermostat has a house code address for X-10 communication. The address of the thermostat is changed using the SW1 dipswitch on the control unit.

Both the TX15 and TX15-B thermostats decode the regular X-10 commands. The TX15-B thermostat can also decode X-10 dim commands. These commands are interpreted as group commands. More information about such commands can be found in [11]. Table 3.3 shows one of the X-10 thermostat decode tables. The other decode table can be found in [11].

Table 3.3 Decode table of X-10 commands for TX15-B thermostat.

| ***Device code*** | ***ON Command*** | | ***OFF Command*** | |
|---|---|---|---|---|
| | **ºF** | **ºC** | **ºF** | **ºC** |
| 1 | 72 | 17 | SYSTEM OFF | |
| 2 | 73 | 18 | HEAT MODE | |
| 3 | 74 | 19 | COOL MODE | |
| 4 | 75 | 20 | AUTO MODE | |
| 5 | 76 | 21 | 40 | 5 |
| 6 | 77 | 22 | 60 | 6 |
| 7 | 78 | 23 | 62 | 7 |
| 8 | 79 | 24 | 63 | 8 |
| 9 | 80 | 25 | 64 | 9 |
| 10 | 81 | 26 | 65 | 10 |
| 11 | 82 | 27 | 66 | 11 |
| 12 | 83 | 28 | 67 | 12 |
| 13 | 84 | 29 | 68 | 13 |
| 14 | 86 | 30 | 69 | 14 |
| 15 | 88 | 31 | 70 | 15 |
| 16 | 90 | 32 | 71 | 16 |

Table 3.3 shows how X-10 commands can be used to control the TX15-B thermostat. Note that the temperatures in the Fahrenheit and Celsius columns are not related to one another. The temperatures in the Fahrenheit column are used when the TX15-B thermostat is set to read temperatures in Fahrenheit. When the thermostat is set to read temperatures in Celsius, then commands in the Celsius column are used. For example, to change the set point of the thermostat to 80ºF will require the user to send the

house code that matches the same house code set for the thermostat, e.g., A, then the device code "9", and finally the function ON. To send the same signal from the CM11A interface, one would need to send the combination of house code, device code, house code, and function code. Therefore to program the same set point using a CM11A interface, one would send "A9" followed by "A" and the ON function code.

To make sure that the sample application works with the X-10 thermostat, the X10v6 virtual instrument has to be modified so that the command in Table 3.3 is addressed. The front panel of the X10v6 virtual instrument is shown in Figure 3.8.

The virtual instrument shown is capable of turning a device on or off. To make this virtual instrument able to use with the X-10 thermostat, the device code selector and the ON/OFF switch have to be changed. The changes that need to be changed are shown in Table 3.3. By making the necessary changes, the front panel of X10v6 virtual instrument becomes the front panel shown in Figure 3.9. The block diagram of the virtual instrument is in Appendix A.

Appendix B shows the block diagrams used for publishing the sample application on the Internet. The victl virtual instrument is used to open or close a virtual instrument when used remotely. The ctlcgi2 virtual instrument is used to change the control settings of the sample application through a web browser.

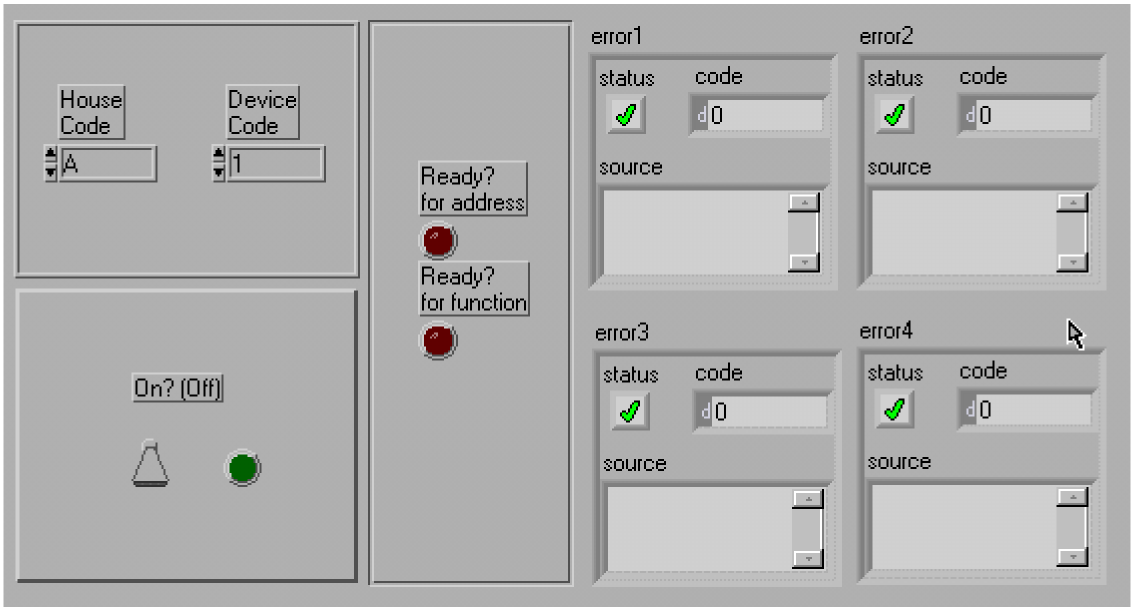



Figure 3.8 Front panel of X10v6 Virtual Instrument.

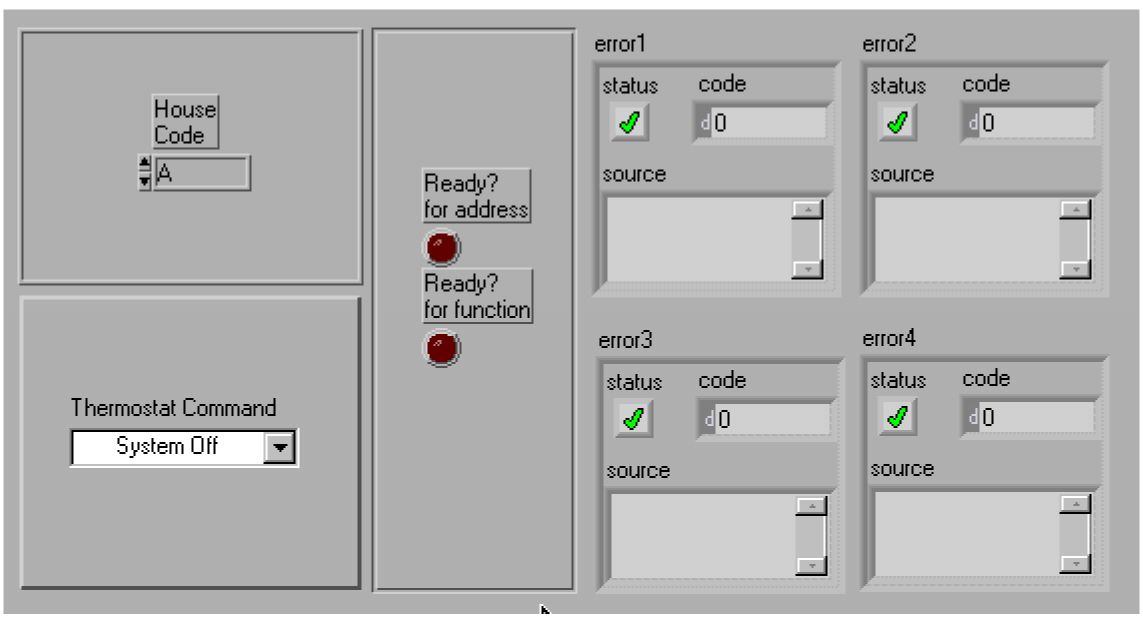



Figure 3.9 Front panel of X10-thermo Virtual Instrument.

# CHAPTER IV

# MODELING AND SIMULATION

## 4.1 Model

In Chapter III, the control of home thermostat is shown to be remotely controllable. However, this does not describe how the system will save on operating costs. Therefore an analytical study is needed to learn more about the system.

The “two resistors and one capacitor” (2R1C) model for building model is quite common in simulating the thermal mass of the air inside a building. Previous work has been done to determine the parameters for models of a town house in New Jersey [12] and a gas-heated house in Ontario [13].

The energy balance equations for the 2R1C model are as follows:

$$Q = H_s(T_i - T_s) + H(T_i - T_o), \tag{4-1}$$

$$C\frac{dT_s}{dt} = H_s(T_i - T_s), \tag{4-2}$$

where:

| | | |
|---|---|---|
| Q | = | sum of all the heat gains within the house [Watts], |
| $T_i$ | = | the average indoor air temperature [°C], |
| $T_s$ | = | the house structure temperature [°C], |
| $T_o$ | = | the outdoor temperature [°C], |
| H | = | the equivalent heat transfer constant between room air and outdoors [Watts/°C], |

$H_s$ = the equivalent heat transfer constant between room air and the house structure [Watts/°C],

C = the equivalent thermal mass of the house [kWh/°C].

The equivalent circuit is shown in Figure 4.1. Note that temperature differences are represented with voltage differences, heat flows with currents, heat conductances by inverse resistances, and the thermal mass with a capacitor. From Figure 4.1, it clearly shows that the model has two resistances and a capacitor, hence the name 2R1C model. Also note that from [12], a *constant temperature clamp* is available in equation (4-1), however, it is ignored here to simplify the system. The model also ignores all other heat inputs, for instance solar heat, indoor humidity, and outside weather variables such as wind.

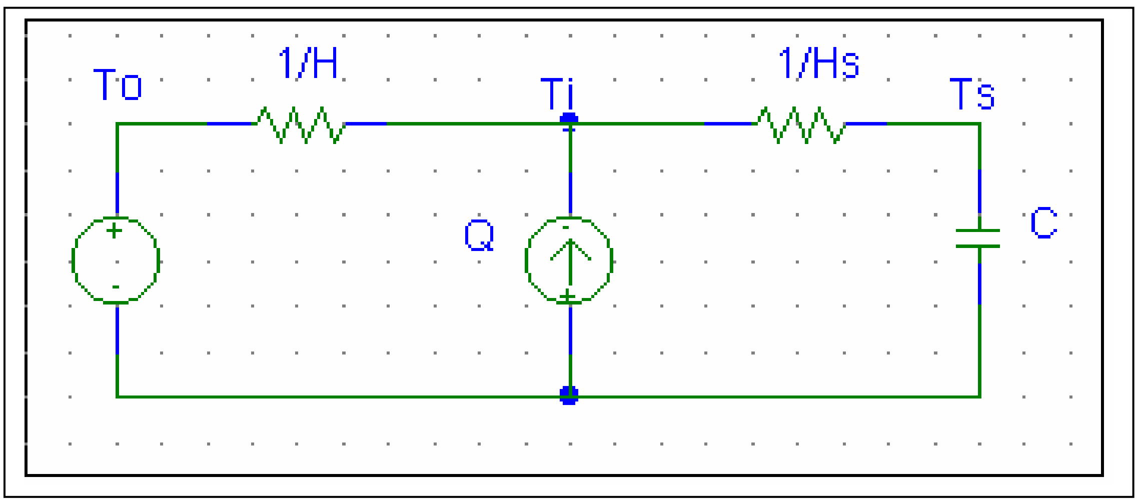



Figure 4.1 Equivalent circuit for the 2R1C model.

The overall system is modeled as a simple nonlinear feedback control system with a *bang-bang* controller as the house heating and cooling (HC) system and part of the 2R1C model as the sensor. In order to combine the two systems, the 2R1C model has to be modified as follows:

$$T_s = \int \frac{dT_s}{dt} dt + T_s(0) \tag{4-3}$$

The Equation (4-2) is rewritten as

$$\frac{dT_s}{dt} = \frac{H_s}{C}(T_i - T_s) \tag{4-4}$$

Next, the sum of heat gains is rewritten as

$$Q = H(T_i - T_o) + C\frac{dT_s}{dt} \tag{4-5}$$

Once the 2R1C model is rewritten, the house HC system can be defined. Note that the 2R1C model is defined for a heating system, however for a cooling system, the sum of all heat gains, Q, will be negative. We will also assume that the initial condition for $T_s$ to be the set point temperature.

The other model that needs to be included is the electrical use of the HC system itself. The electrical use of the system is needed to determine if any *smart* system implemented to the thermostat will save the use of electricity. Most of the HC system can be represented with a *bang-bang* controller since it only turns on or off depending on the set point. From Equation (4-5), Q is obtained from the indoor temperature and the heat flow of the structure temperature. However, since the *bang-bang* controller will act as the HC system, another heat gain is available. Hence, Equations (4-4) and (4-5) are rewritten so that the indoor temperature can be obtained from the heat flow of the structure temperature and the heat gain supply.

From Equation (4-4), $T_i$ can be obtained using

$$T_i = T_s + \frac{C}{H_s}\frac{dT_s}{dt} \tag{4-6}$$

$T_i$ can also be obtained from Equation (4-1) as shown,

$$Q = H_s T_i - H_s T_s + HT_i - HT_o$$

$$Q + H_s T_s + HT_o = T_i\left(H_s + H\right)$$

$$T_i = \frac{Q + H_s T_s + HT_o}{H_s + H} \tag{4-7}$$

Figures 4.2 – 4.5 illustrate the overall model of the system with the nonlinear controller as the HC system and part of the 2R1C model as the sensor. Note that the bang-bang controller is slightly modified. When a cooling system is used, the bang-bang controller has a negative unity offset, but when a heating system is used, the bang-bang controller has a positive unity offset. This is due to the fact that cooling is represented with a negative heat gain while heating is represented with a positive heat gain. The heating or cooling system is assumed to be able to provide up to 6kW of heating or cooling power. The presence of the limiter in the 2R1C model is just to make sure that the calculated heat gain is limited to 6. Tables 4.1 and 4.2 show the parameters used for the simulating both cooling and heating systems.

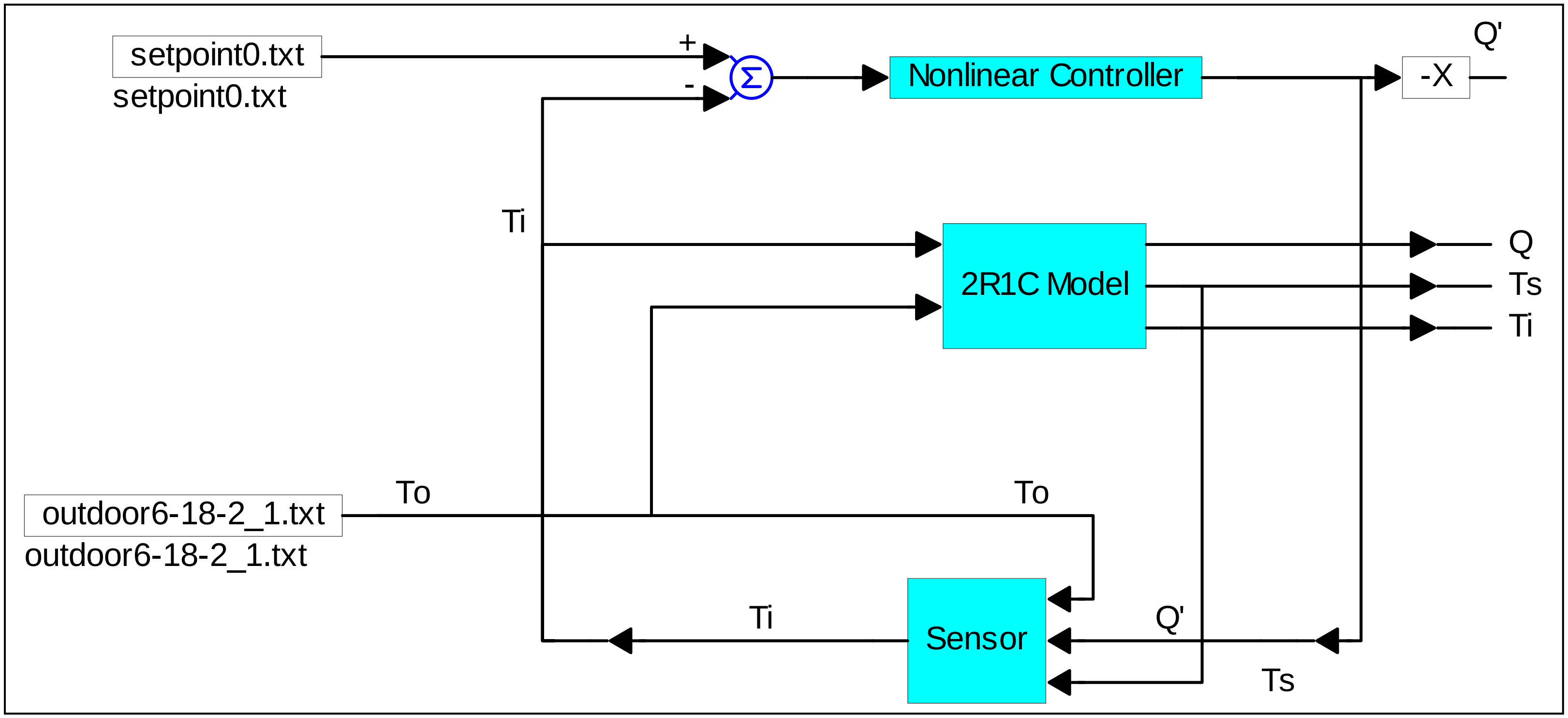



Figure 4.2 The overall model used for simulation.

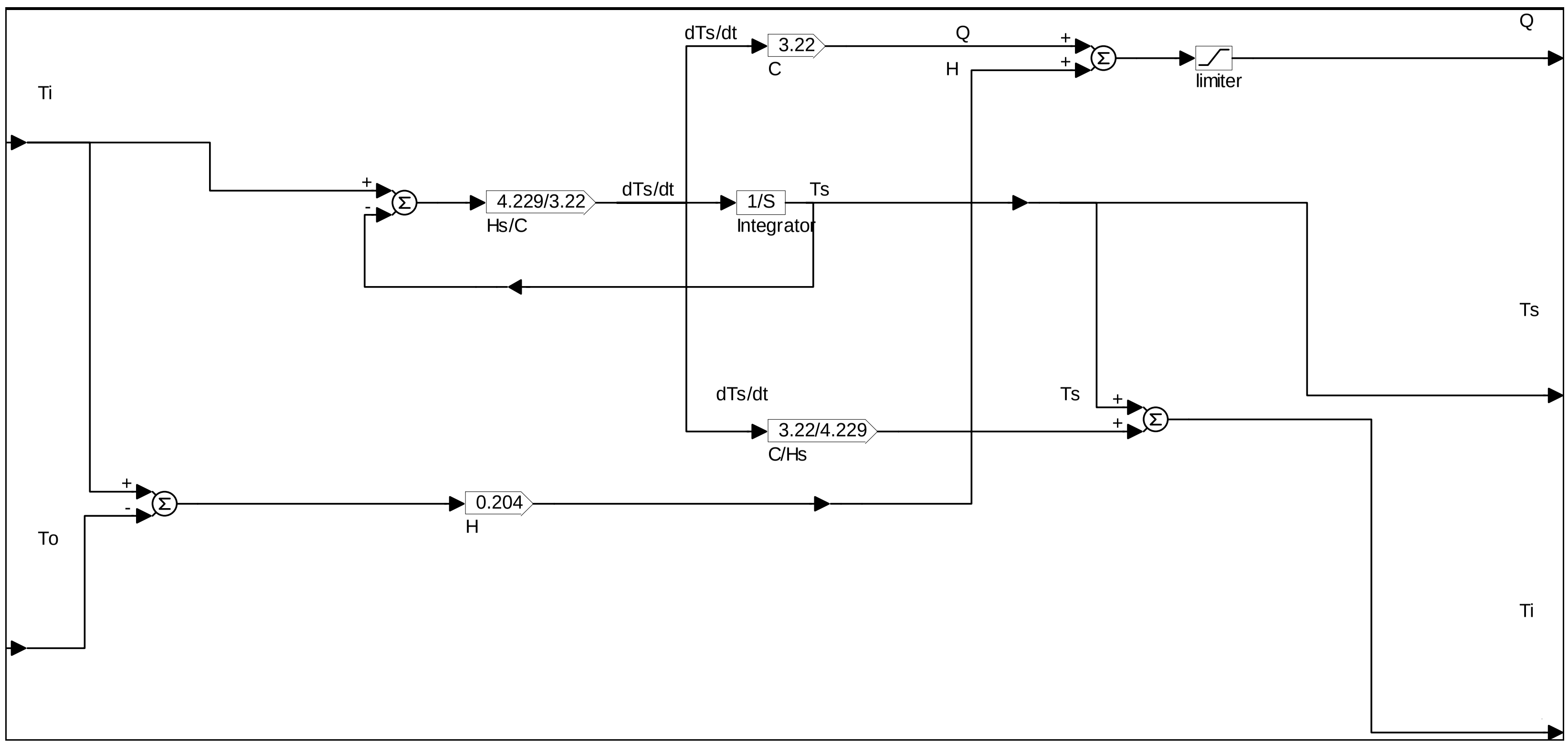



Figure 4.3 The 2R1C model used in the overall model.

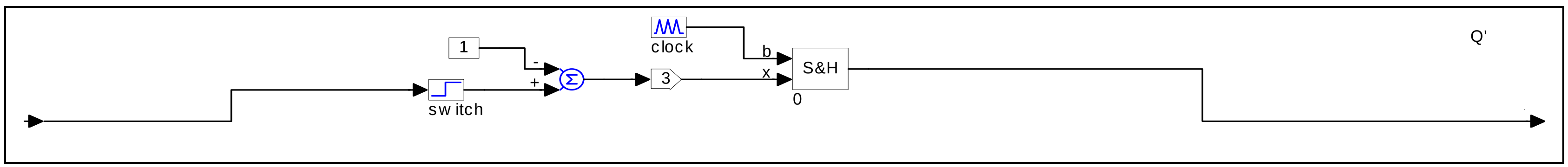



Figure 4.4 The nonlinear bang-bang controller used in the overall model.

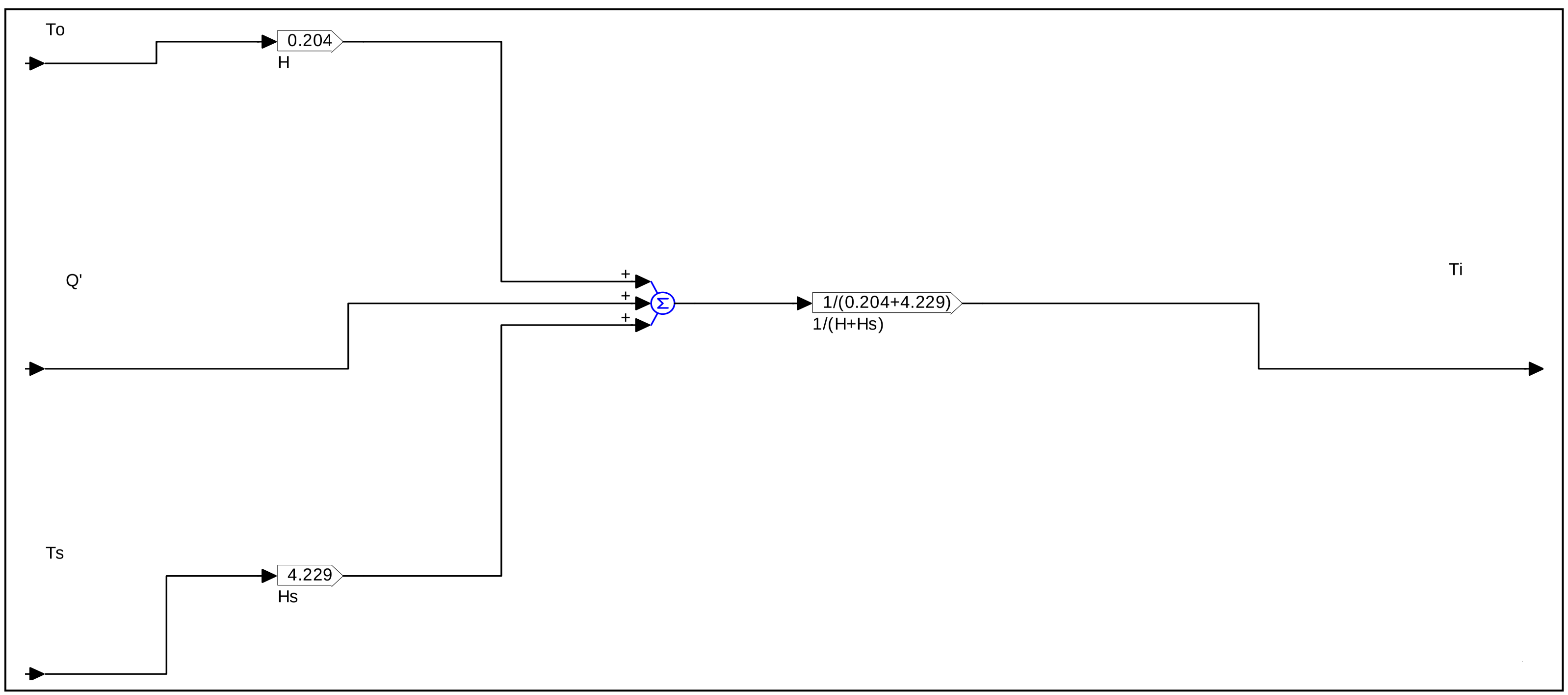



Figure 4.5 The sensor used in the overall model.

Table 4.1 Simulation parameters used for a cooling system.

| **Name** | **Value** |
|---|---|
| H | 0.204 kW/ºC |
| $H_s$ | 4.229 kW/ºC |
| C | 3.22 kWh/ºC |
| $Q_{min}$ | 0.0 kW |
| $Q_{max}$ | 6.0 kW |
| $T_{i(min)}$, $Ts_{(initial)}$ | 21.11ºC |
| $T_{i(max)}$ | 26.67ºC |
| $T_o$ | Refer to Figure 4.6 |

Table 4.2 Simulation parameters used for a heating system.

| **Name** | **Value** |
|---|---|
| H | 0.204 kW/ºC |
| $H_s$ | 4.229 kW/ºC |
| C | 3.22 kWh/ºC |
| $Q_{min}$ | 0.0 kW |
| $Q_{max}$ | 6.0 kW |
| $T_{i(min)}$ | 15.56ºC |
| $T_{i(max)}$, $Ts_{(initial)}$ | 21.11ºC |
| $T_o$ | Refer to Figure 4.7 |

### 4.2 Simulation results and analysis

The input for the simulation set point is a pulse train consisting of maximum set point and minimum set point values. The set point is used to simulate the temperature set by the user hourly. For the cooling system simulation, the set point is 21.11ºC (70ºF) when there are people in the house, for example from 4pm to 7am, and to 26.67ºC (80ºF) when there are no people in the house, for example from 8am to 3pm. On the other hand, the heating system simulation has set point of 21.11ºC (70ºF) when the house is occupied and 15.56ºC (60ºF) when the house is not occupied. The outdoor temperature is obtained from [14] daily from June 18 2001 to June 22 2001 for the cooling system and from November 26 2002 to November 29 2002 for the heating system. Note that these temperatures are updated everyday. However, the data used for simulation can be found in Appendix C. The simulation model is simulated using real world data to imitate the electricity use of a home HC system. Figures 4.6 and 4.7 show the outdoor temperature and the set point used in the simulation.

The results of the simulations are displayed in Figures 4.8 and 4.10. From the results, the number of hours that the HC system is on has to be determined. Apart from that, we need to compare the results with a non-changing set point simulation. The second simulation will imitate the system that most people use with their heating and cooling system at home, set it at a comfortable temperature and leave it there. In this case, the set point temperature is set to 21.11ºC. The results of the simulation are shown in Figures 4.9 and 4.11. Again, the number of hours that the HC system used is determined.

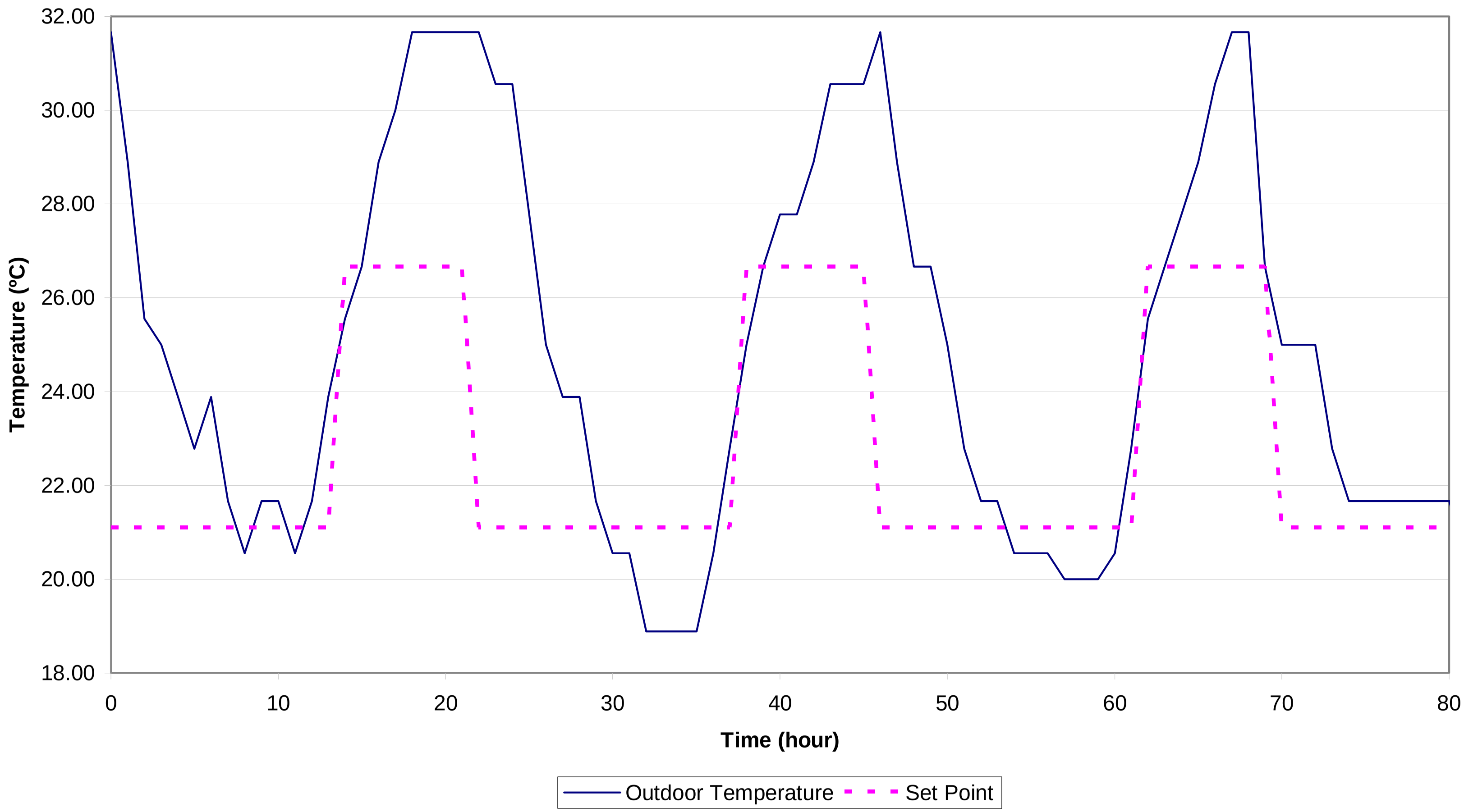



Figure 4.6 Outdoor temperature and set point used in cooling system simulation.

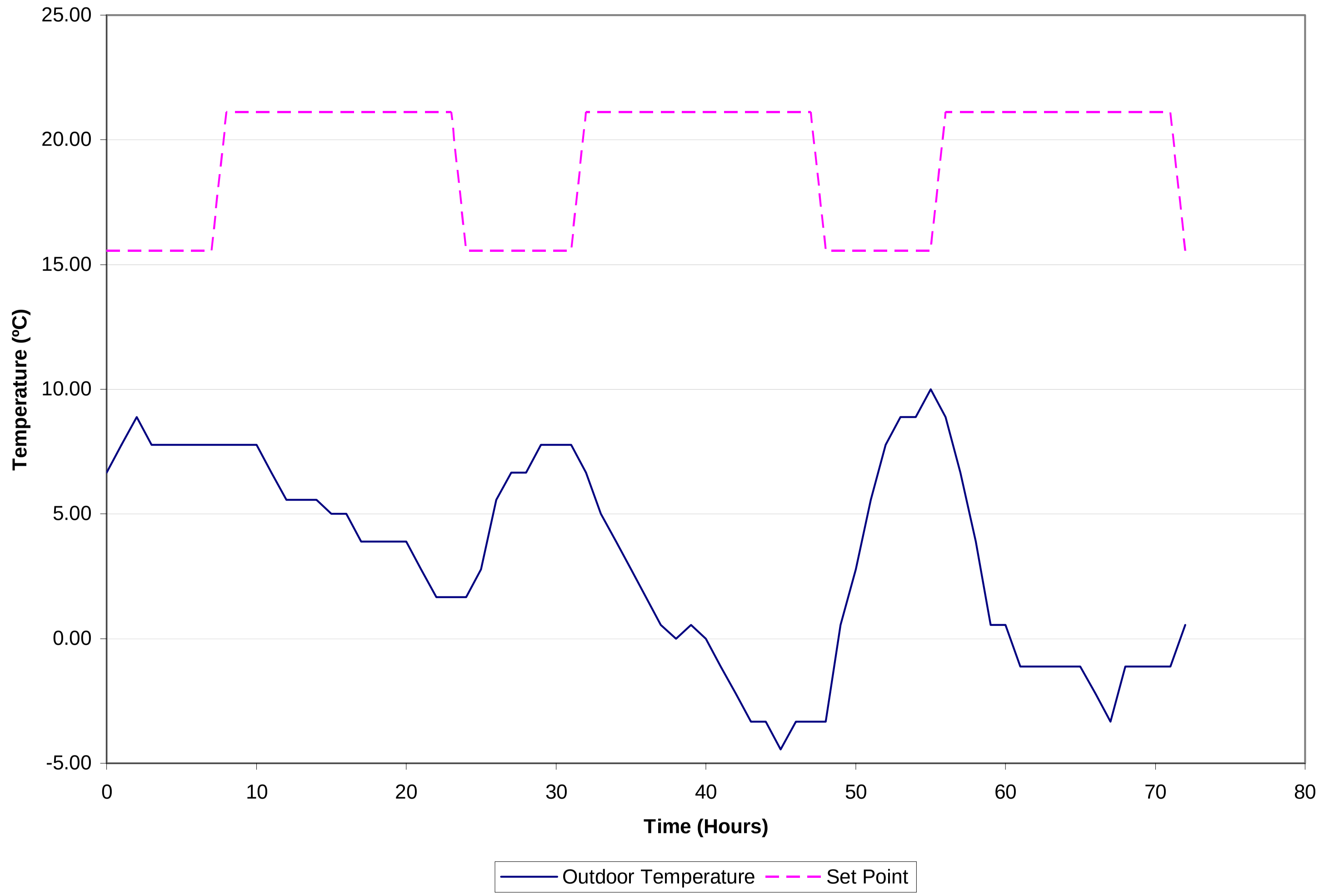



Figure 4.7 Outdoor temperature and set point used in heating system simulation.

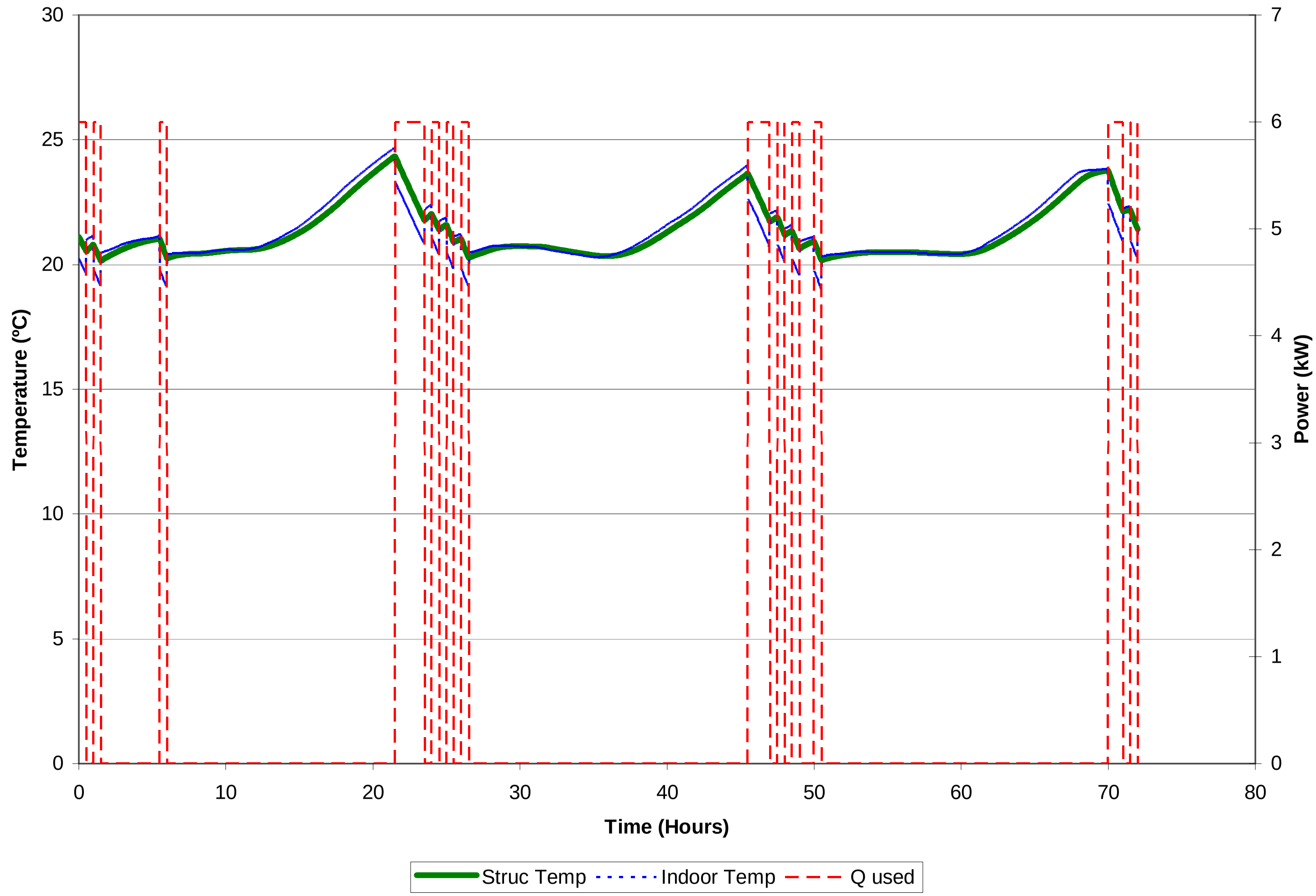



Figure 4.8 Cooling system simulation results with changing set point.

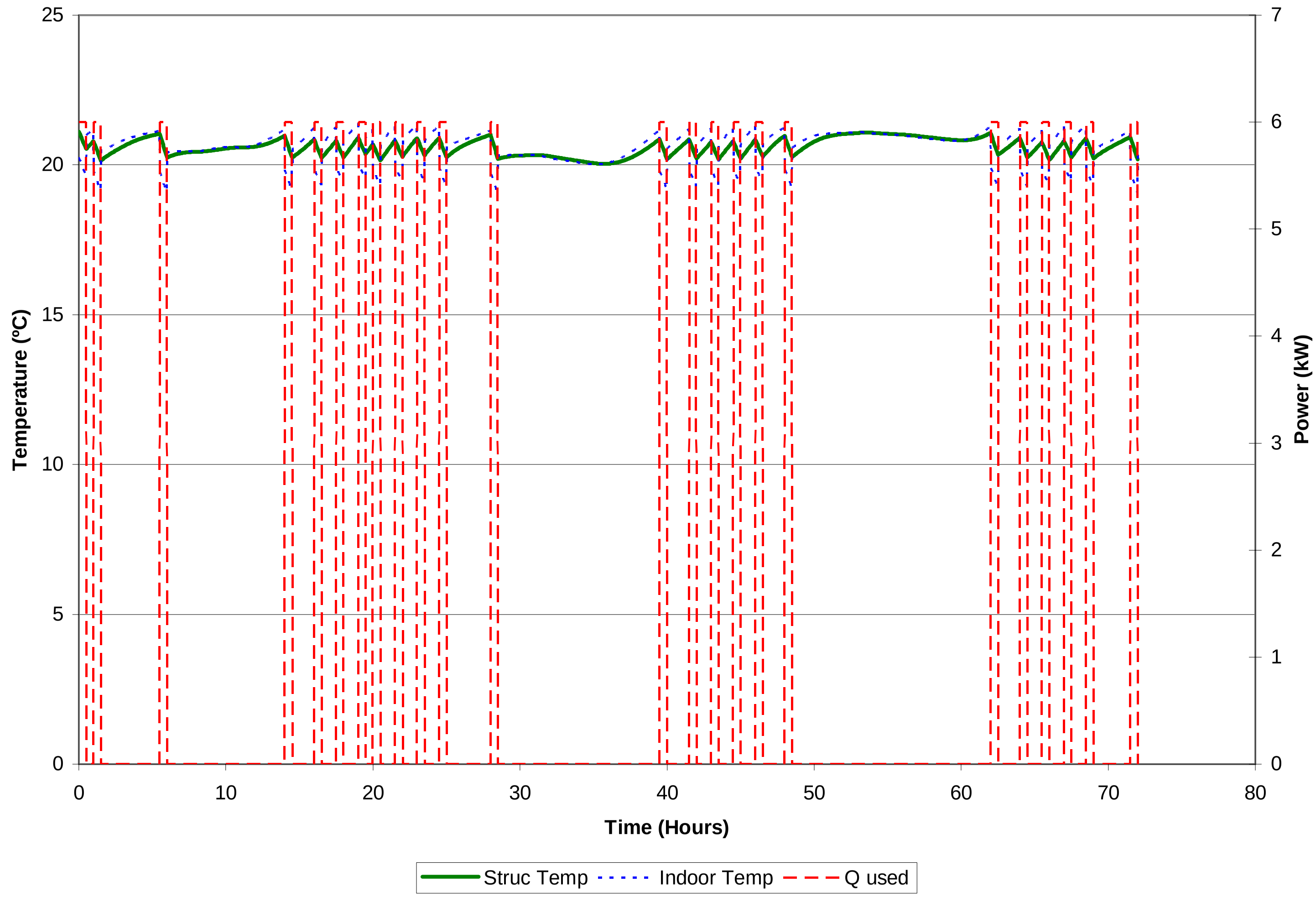



Figure 4.9 Cooling system simulation results with fixed set point.

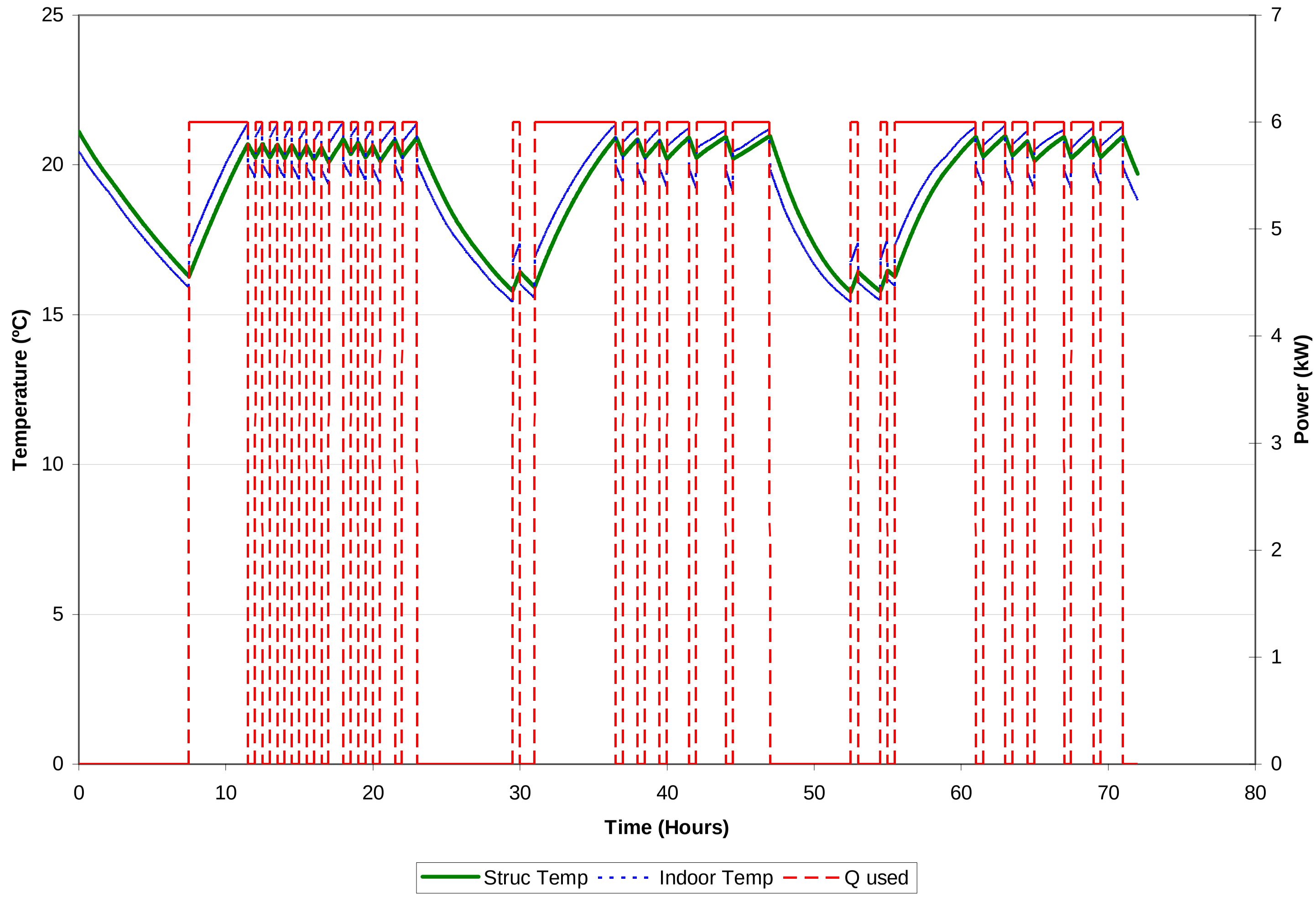



Figure 4.10 Heating system simulation results with changing set point.

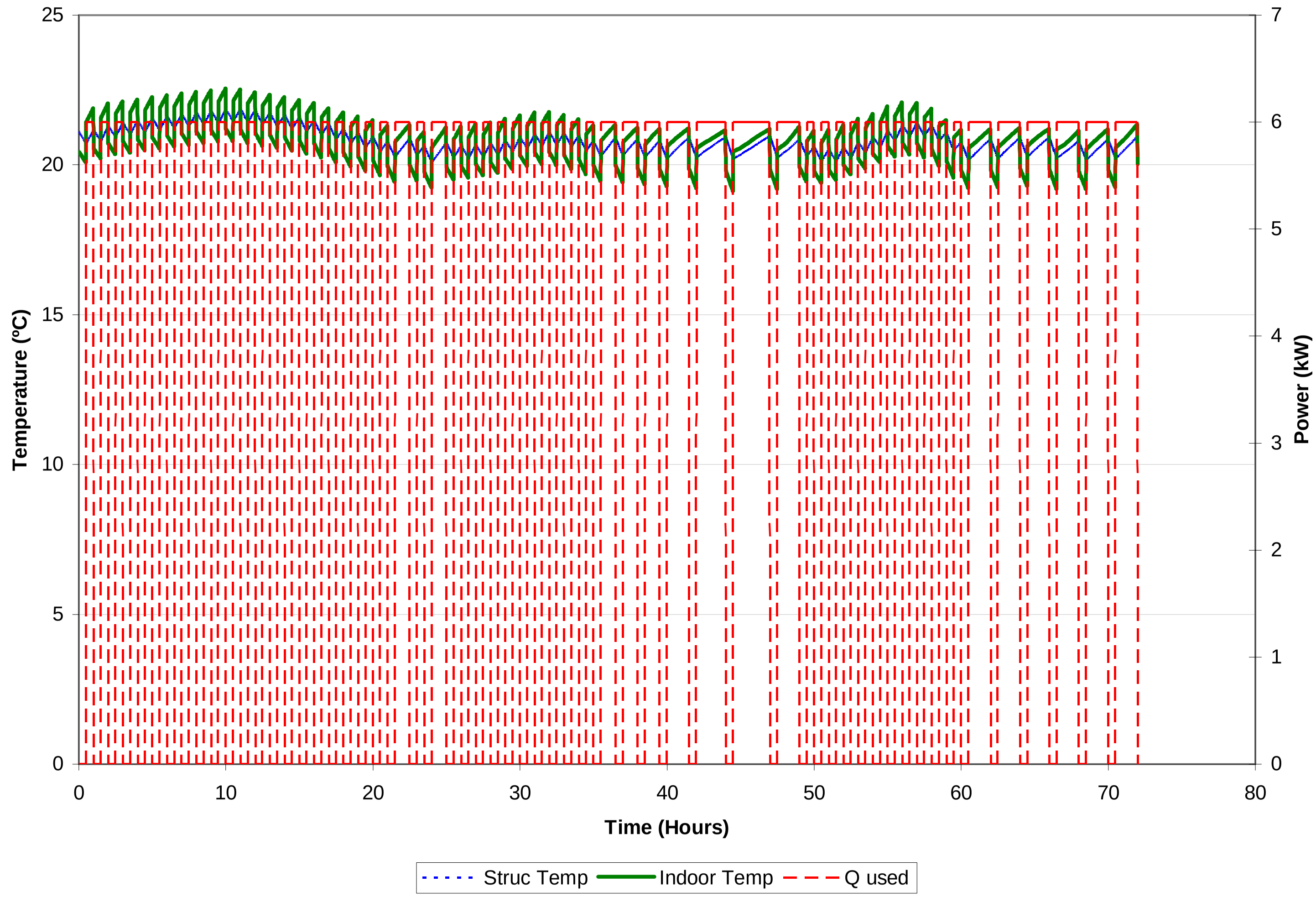



Figure 4.11 Heating system simulation results with fixed set point.

From Figure 4.9, the cooling system with a fixed set point is turned on more often than the simulation results found in Figure 4.8 with a variable set point. The same is true for the heating system simulation with the fixed set point system on more often than the variable set point. Table 4.3 shows the number of hours the HC system that was turned on in the simulations. The yearly calculations are based on nine months of using the cooling system and three months of using the heating system. When using the X-10 thermostat, the user might save about 2100kWh of electricity use yearly. The estimated annual savings is about $140.

Table 4.3 Estimated electrical usage based on HC system.

| **Simulation** | **Number of hours HC system turned on for a three-day period** | **Electrical usage per month (kWh) (based on HC alone)** | **Estimated monthly cost (based on $0.07/kWh)** | **Electrical usage per year (kWh) (based on HC alone)** | **Estimated yearly cost (based on $0.07/kWh)** |
|---|---|---|---|---|---|
| Changing set point (Cooling) | 9.45 | 567 | $39.69 | 12033 | $842.31 |
| Changing set point (Heating) | 38.5 | 2310 | $161.70 | | |
| Fixed set point (Cooling) | 11.79 | 707.4 | $49.52 | 14106.6 | $987.46 |
| Fixed set point (Heating) | 43 | 2580 | $180.60 | | |

Table 4.4 shows the results obtained from simulation using data collected hourly for year 2002. The hourly climatological data can be obtained as a subscription from [15]. The simulation is done by splitting all the data into month segments. Since the simulation can only be either a heating system or a cooling system, the monthly data has to be divided into heating and cooling segments. The set point data to the simulation is determined by the outdoor temperature. If the outdoor temperature is more than 21.11ºC during the hours that the house is not occupied, the set point is set to 25.56ºC and 15.56ºC if the outdoor temperature is less than 21.11ºC. The fixed set point simulation is done by assuming that the system will be a heating system whenever the outdoor temperature is about 21.11ºC, and a cooling system otherwise.

Table 4.4 Estimated electrical usage using data from year 2002.

| **Simulation** | **Electrical usage per year (kWh) (based on HC alone)** | | **Estimated yearly cost (based on $0.07/kWh)** | |
|---|---|---|---|---|
| | Heating | Cooling | Heating | Cooling |
| Changing set point | 8484.90 | 3490.08 | $593.94 | $244.31 |
| Fixed set point | 10135.02 | 4699.26 | $709.45 | $328.95 |

Table 4.4 shows that heating uses about twice as much energy as cooling does. Table 4.4 also shows that the user can save about 2900kWh of total heating and cooling energy, which is equivalent to about $200 of savings per year. It clearly shows us that using a programmable thermostat like the X-10 thermostat can reduce energy consumption, which translates in financial savings for the user.

### 4.3 Conclusion

Based on the results shown in this chapter, it is clear that by using a *smart* thermostat, we use less electricity than with a normal thermostat. Using a regular programmable thermostat can save money, however, one might not remember to pre-program settings before leaving for a long vacation. Hence the programmable thermostat becomes a regular thermostat. However, a remote-controllable thermostat is more useful with the capability of logging in to the X-10 server to change the settings at any time in the day.

# CHAPTER V

# CONCLUSION AND FUTURE WORK

The inner workings of X-10 have been examined so that a good understanding of how this home automation standard works is obtained. From the simulation results, a scheduled HVAC system can save energy. The X-10 system used is mainly for remote control of such HVAC system. A combination of the two systems proves to be a good combination, especially if the system is used during summer vacations.

One can argue that all the HVAC system can be turned off when going for a long vacation and the system can be turned back on when one arrives home. Probably this is what most people have been doing all this time. However, it would be more comfortable to return home at a comfortable temperature, rather than at the same temperature as the outdoors.

Most people will just set their thermostat to a certain set point, and leave the settings for the most part of the year. If one would change the thermostat daily to save electricity consumption, it might not work all the time. One has to remember to change the setting daily. If this is not done, then it will be the same as leaving the thermostat with a static set point. Therefore, a dynamic thermostat set point using a PC could help tremendously.

Apart from that the ability to control the schedule through the Internet will be very helpful too. This is especially useful if one is not at home for a long period of time, and would like to change the thermostat set point remotely.

Additional systems can be added to the current configuration. A security system using X-10 devices is useful too when one is not at home for a long period of time. An emergency routine might also be implemented so that any emergency can be reported to the authorities. This would be helpful to prevent any kind of emergencies to a house.

# REFERENCES


[1] Wacks, K. P., "International development of home automation standards," Digest of technical papers, International Conference on Consumer Electronics, 1992.

[2] Numes, R., "An architecture for a home automation system," Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems, Sept. 1998.

[3] http://www.echelon.com/Products/Core/faq.htm

[4] Introduction to the LonWorks System, IntroLonWorks.pdf

[5] Douligeris, C., "Intelligent home systems," IEEE Communications Magazine, Oct. 1993.

[6] http://www.geocities.com/ido_bartana/x10theor.htm

[7] Paton, B., Sensors, Tranducers & LabView, Prentice Hall, Upper Saddle River, NJ, 1999.

[8] Jeffrey, T., Internet applications in LabView, Prentice Hall, Upper Saddle River, NJ, 2000.

[9] Omega, The temperature handbook: 21st century preview edition, Omega, 1998.

[10] CM11A programming protocol, ftp://ftp.x10.com/pub/manuals/cm11a_protocol.txt.

[11] Model TX15-B thermostat X-10 bi-directional protocol manual.

[12] Sonderegger, R. C., "Diagnostic tests determining the thermal response of a house," ASHREA Transactions, Vol. 84 Pt. 1, 1978, pp. 691-702.

[13] Wilson, N. W., Wagner, B. S., and Colborne, W.G., "Equivalent thermal parameters for a gas-heated house," ASHREA Transactions, Vol. 91 Pt. 2, 1985, pp. 1875-1885.

[14] http://weather.noaa.gov/weather/current/KGTR.html

[15] http://nndc.noaa.gov/?http://ols.nndc.noaa.gov/plolstore/plsql/olstore.prodspecific?prodnum=C00128-PUB-S0001

# APPENDIX A

# VIRTUAL INSTRUMENT SOURCE CODE

The following VI's are used to imitate an actual HVAC system. For each VI, connector pane, front panel, block diagram, and position in hierarchy are displayed.

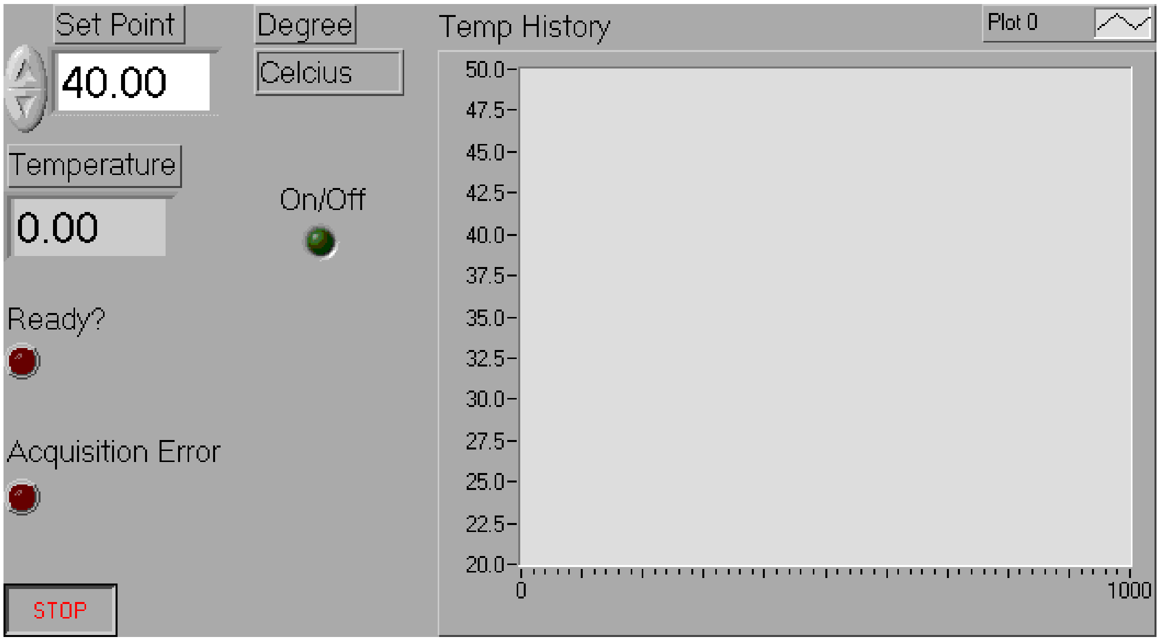
The connector pane shows the input and output of a VI. Inputs to a VI are listed on the left side of the VI icon and outputs are listed on the right side of the icon. If a VI does not have any input or output, then it only accepts user controls. Front panel is the user interface to a VI. All user controls are displayed. Block diagram is the source code for a VI, and position in hierarchy shows all the sub-VI's used. Data in block diagrams flow from left to right. Inputs are normally placed on the left side and outputs on the right side of the block diagram.

## A.1 Controlv1.vi

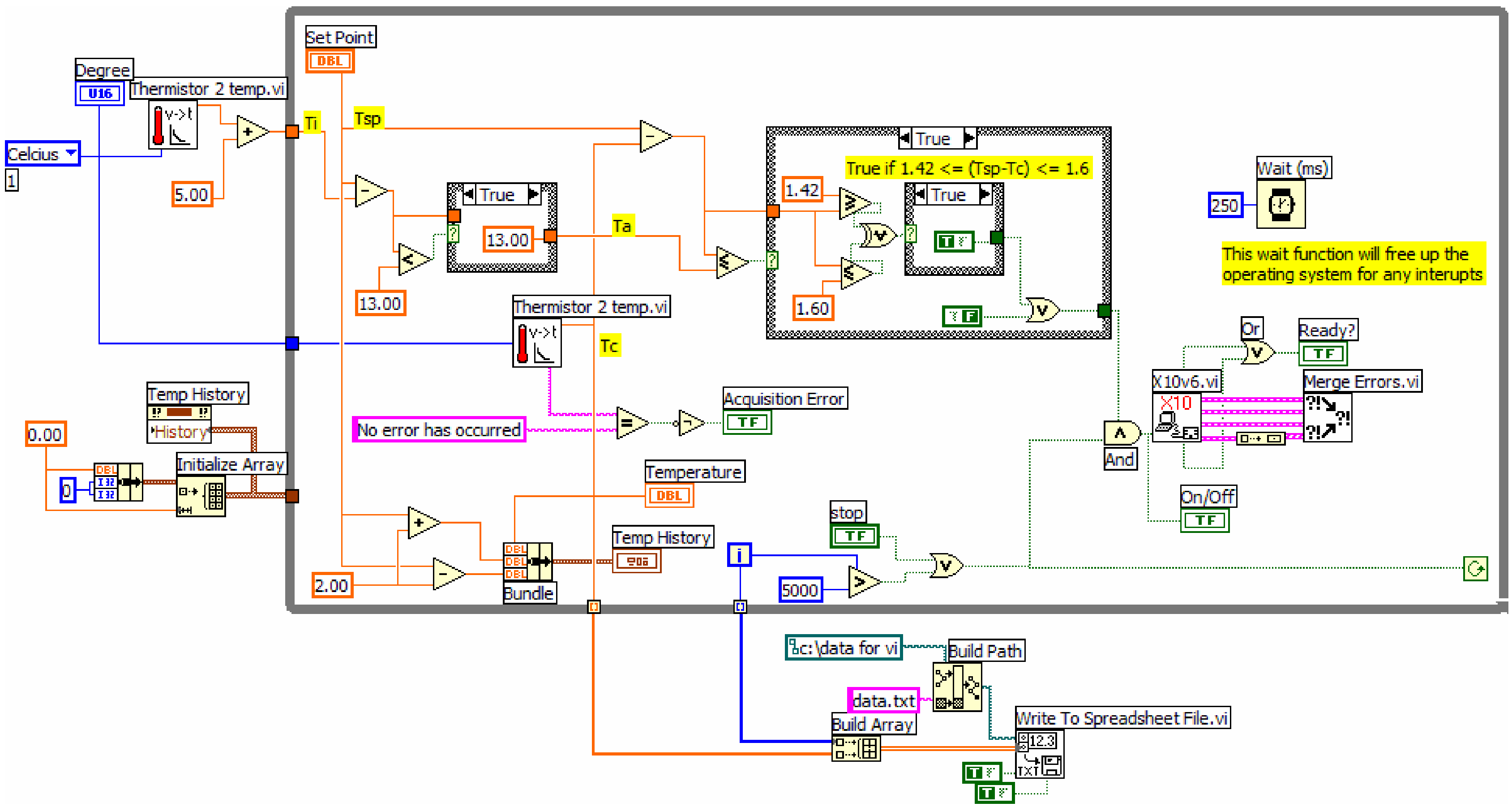
This is the VI used to heat an iron block to the set point temperature.

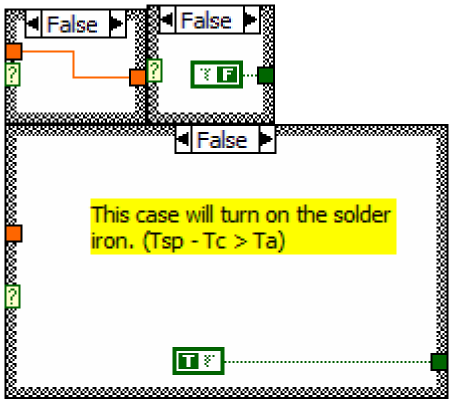
### A.1.1 Controlv1 VI Connector Pane

A.1.2 Controlv1 VI Front Panel

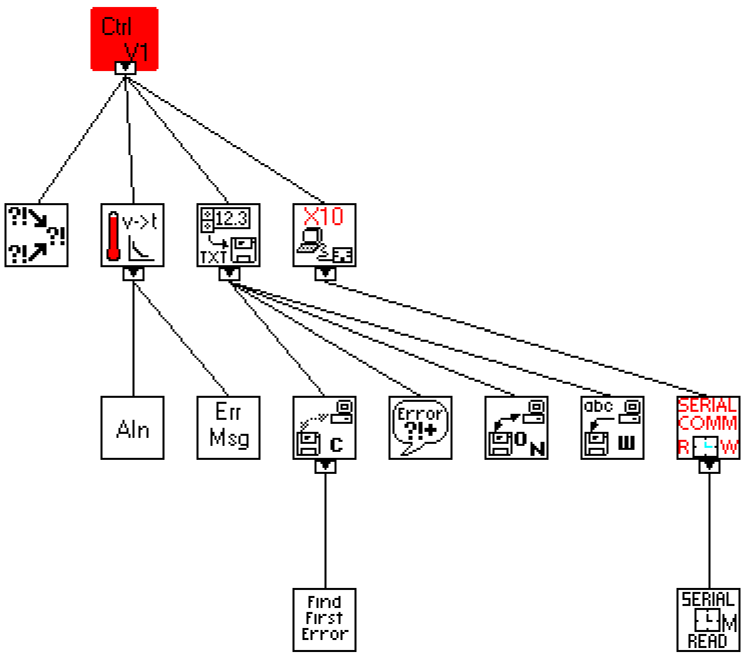


## A.1.3 Controlv1 VI Block Diagram





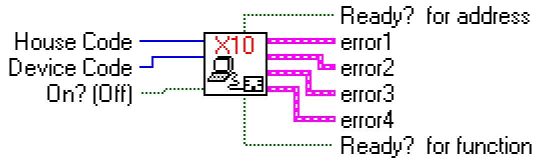

A.1.4 Controlv1 VI Position in Hierarchy



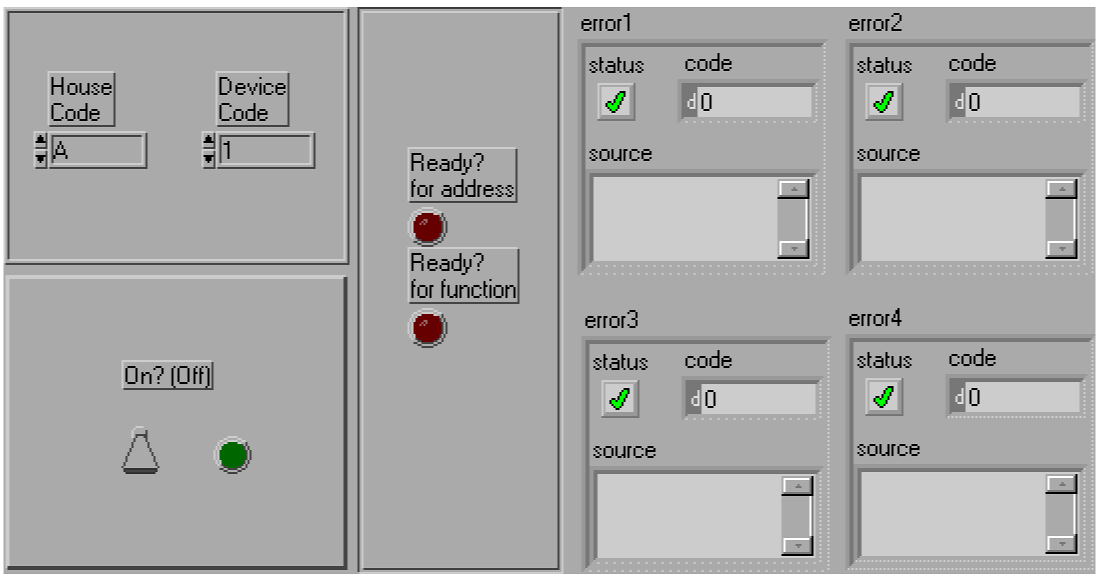

**A.2 X10v6.vi**

This VI communicates with other X-10 through the CM11A module. It sends function codes to selected X-10 devices.

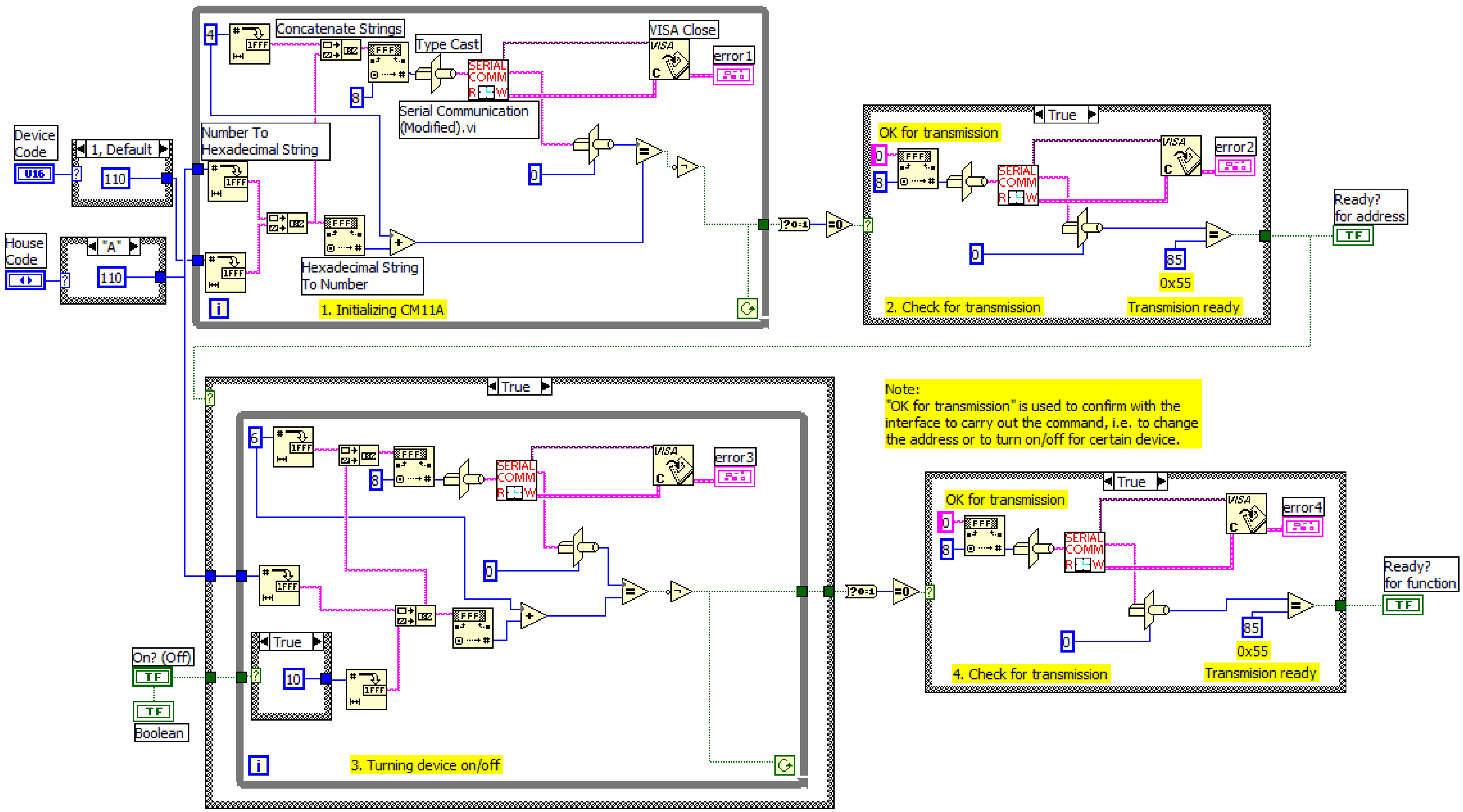
## A.2.1 X10v6 VI Connector Pane

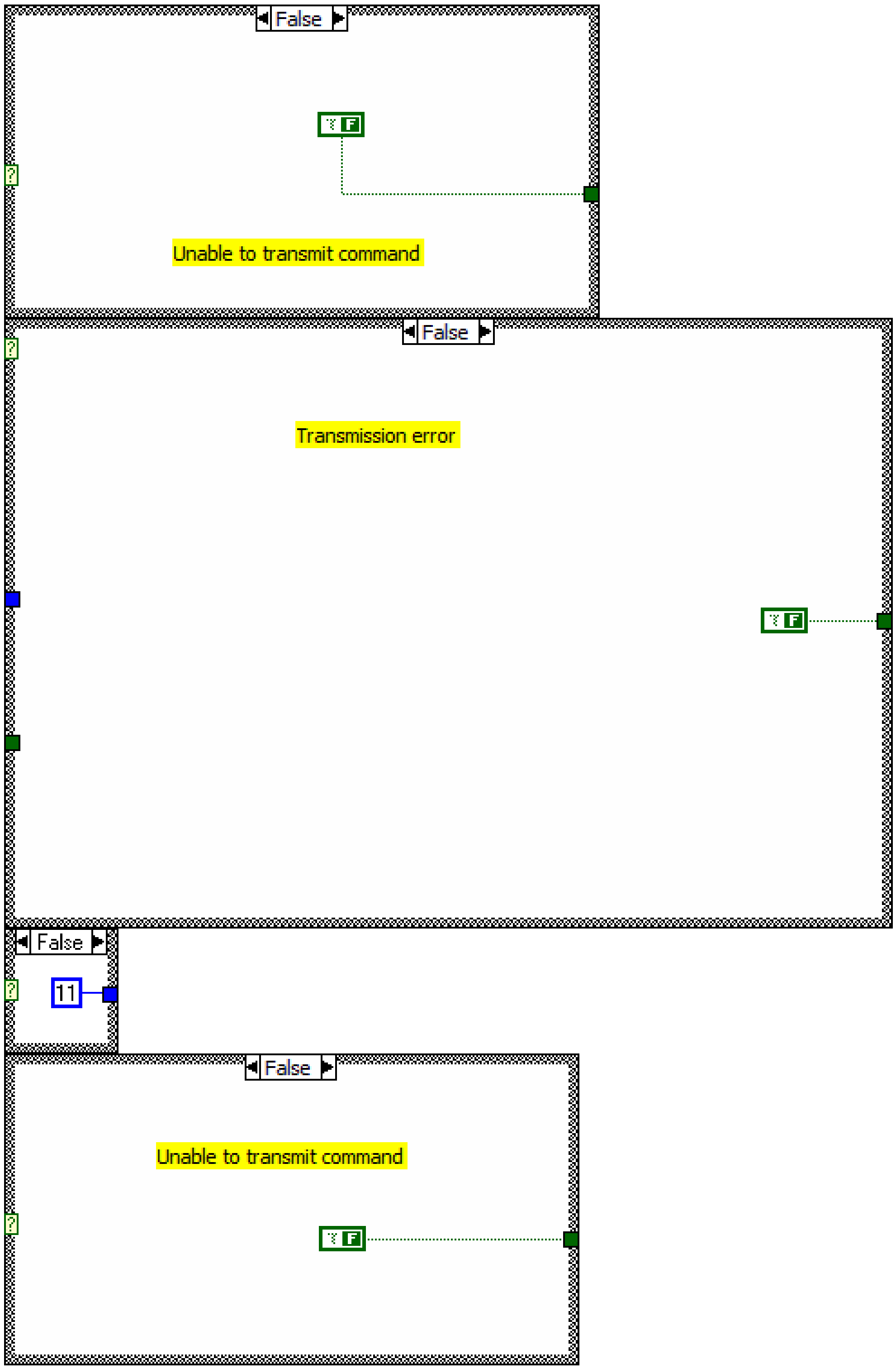


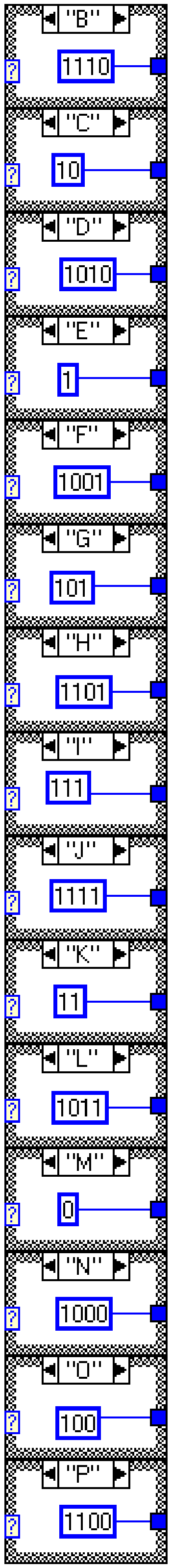

## A.2.2 X10v6 VI Front Panel

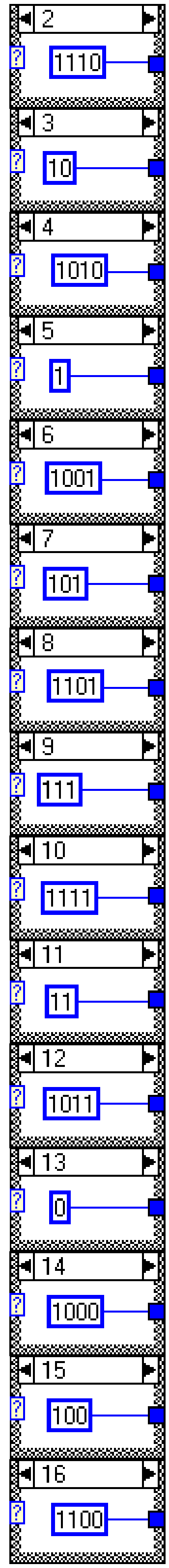


## A.2.3 X10v6 VI Block Diagram

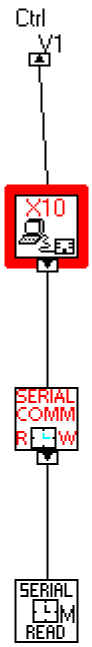



False
T F
Unable to transmit command
False
Transmission error
T F
False
11
False
Unable to transmit command
T F


"B"
1110
"C"
10
"D"
1010
"E"
1
"F"
1001
"G"
101
"H"
1101
"I"
111
"J"
1111
"K"
11
"L"
1011
"M"
0
"N"
1000
"O"
100
"P"
1100


2
1110
3
10
4
1010
5
1
6
1001
7
101
8
1101
9
111
10
1111
11
11
12
1011
13
0
14
1000
15
100
16
1100

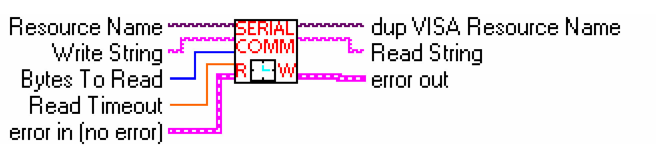
A.2.4 X10v6 VI Position in Hierarchy



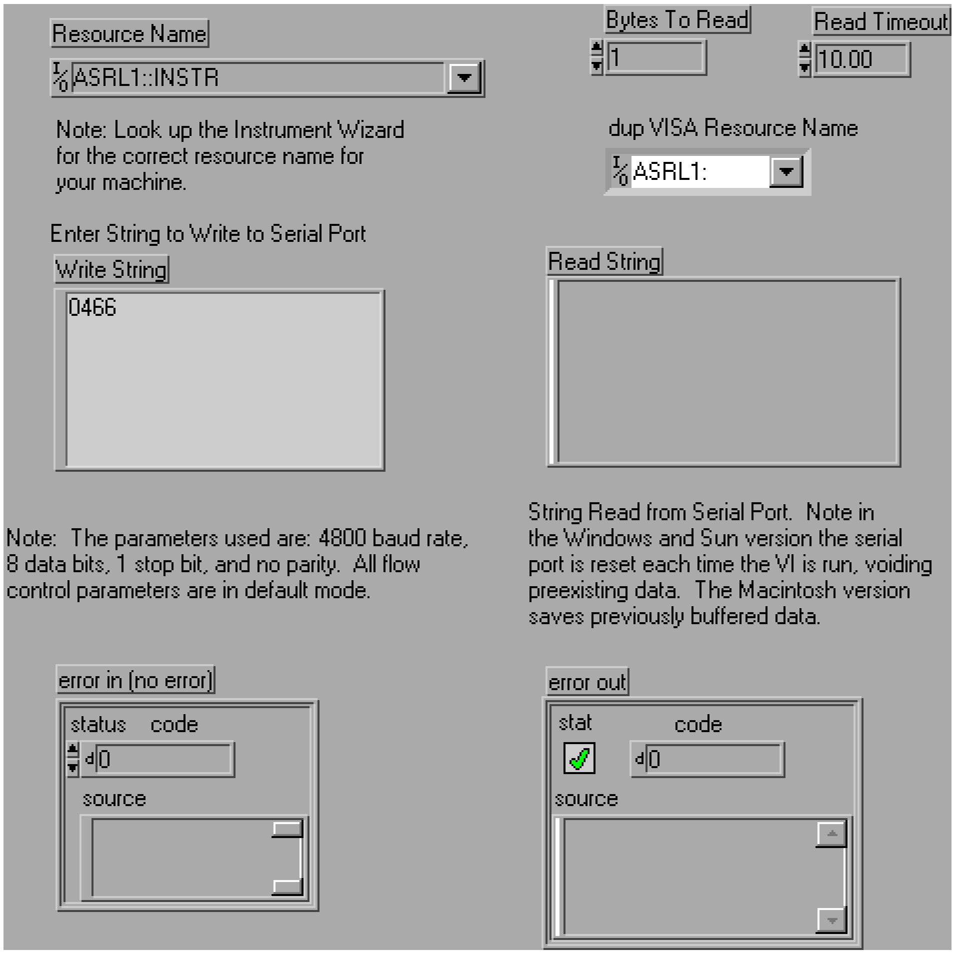

## A.3 Serial Communication (Modified).vi

The Serial Communication (Modified) VI performs bidirectional communication with a serial port. It initializes the port, writes a string to the port, and performs a read with timeout.

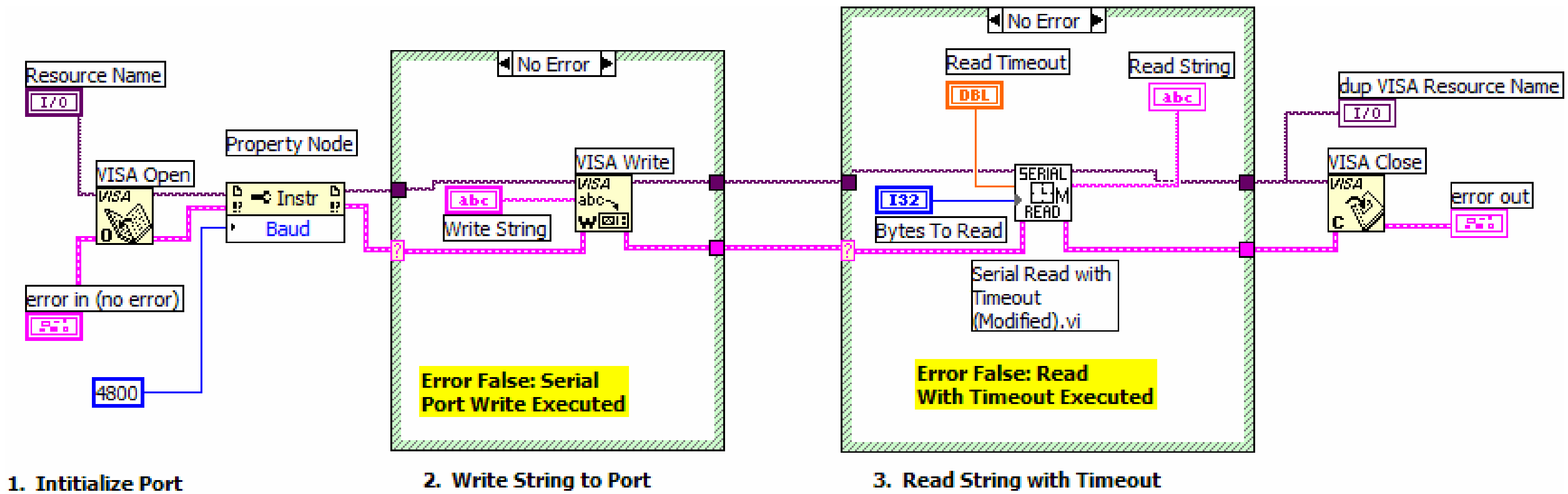
A.3.1 Serial Communication (Modified) VI Connector Pane

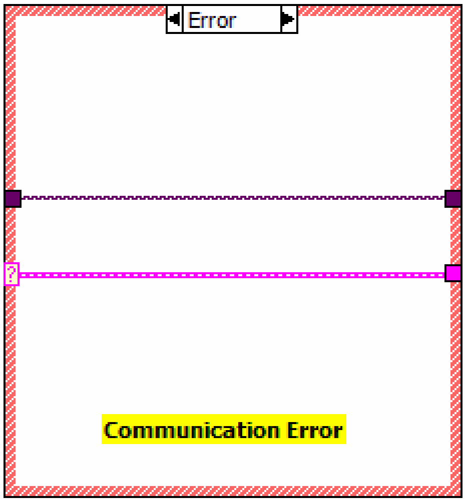


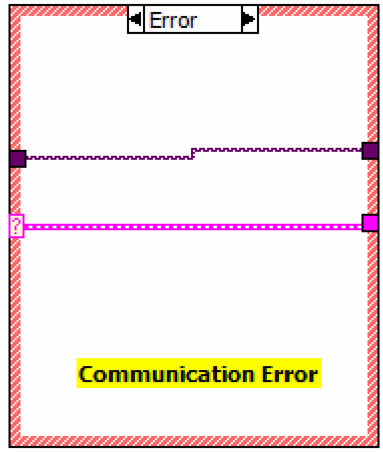
A.3.2 Serial Communication (Modified) VI Front Panel



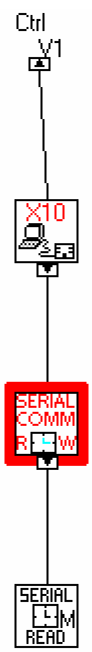
A.3.3 Serial Communication (Modified) VI Block Diagram





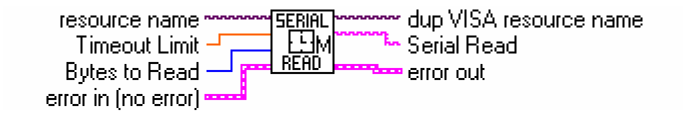

A.3.4 Serial Communication (Modified) VI Position in Hierarchy



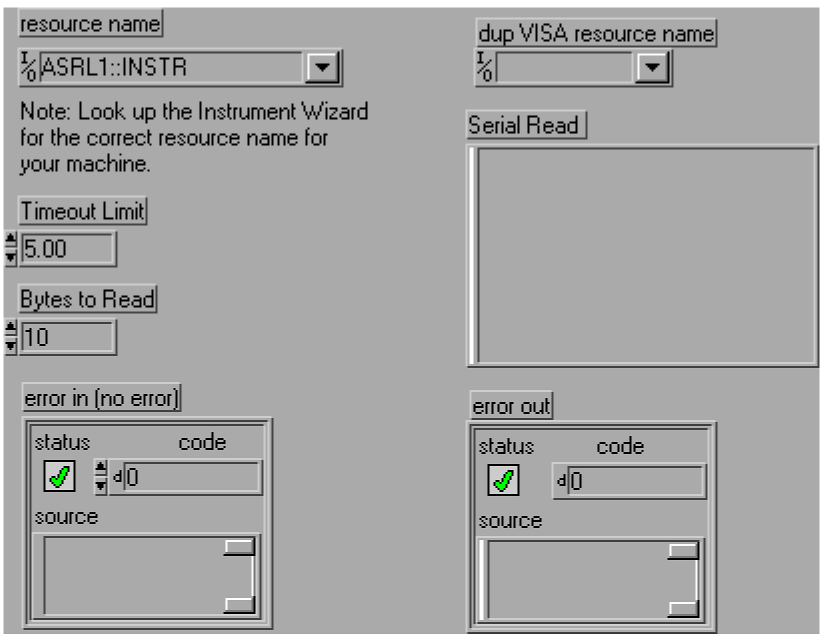
**A.4 Serial Read with Timeout (Modified).vi**

The Serial Read with Timeout (Modified) VI reads bytes of data until either the requested number of bytes is received, or until a timeout occurs.

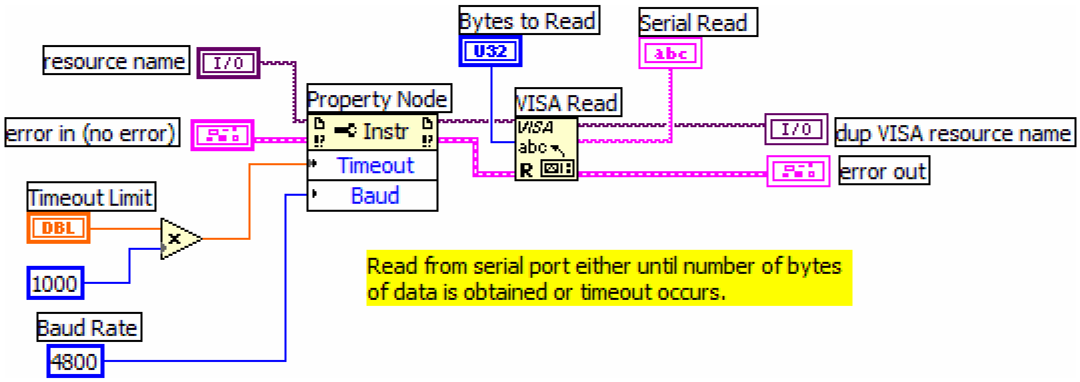
A.4.1 Serial Read with Timeout (Modified) VI Connector Pane



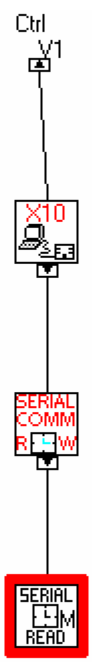

A.4.2 Serial Read with Timeout (Modified) VI Front Panel



A.4.3 Serial Read with Timeout (Modified) VI Block Diagram



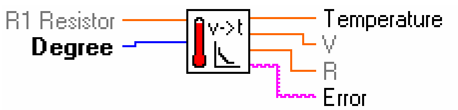

A.4.4 Serial Read with Timeout (Modified) VI Position in Hierarchy



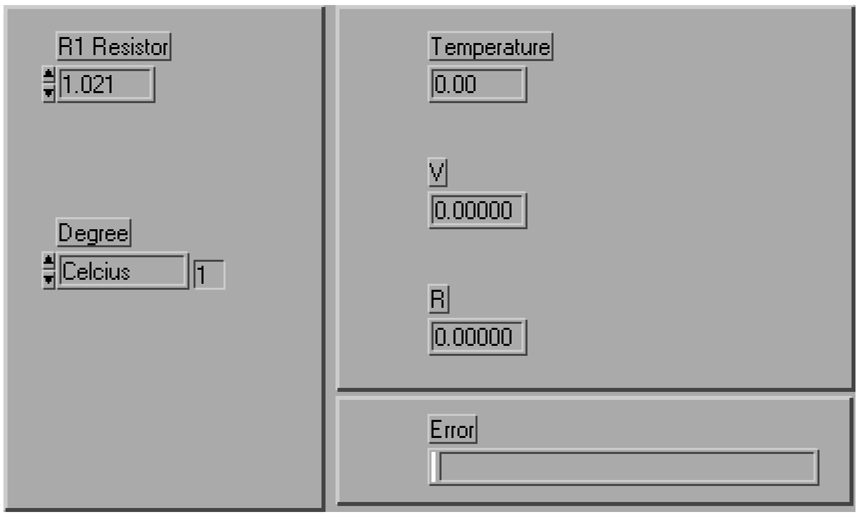

**A.5 Thermistor 2 temp.vi**

This VI converts voltage from a thermistor to temperature. The voltage of the thermistor is measured using a data acquisition card. More information about the setup of the circuit can be found in the main text.

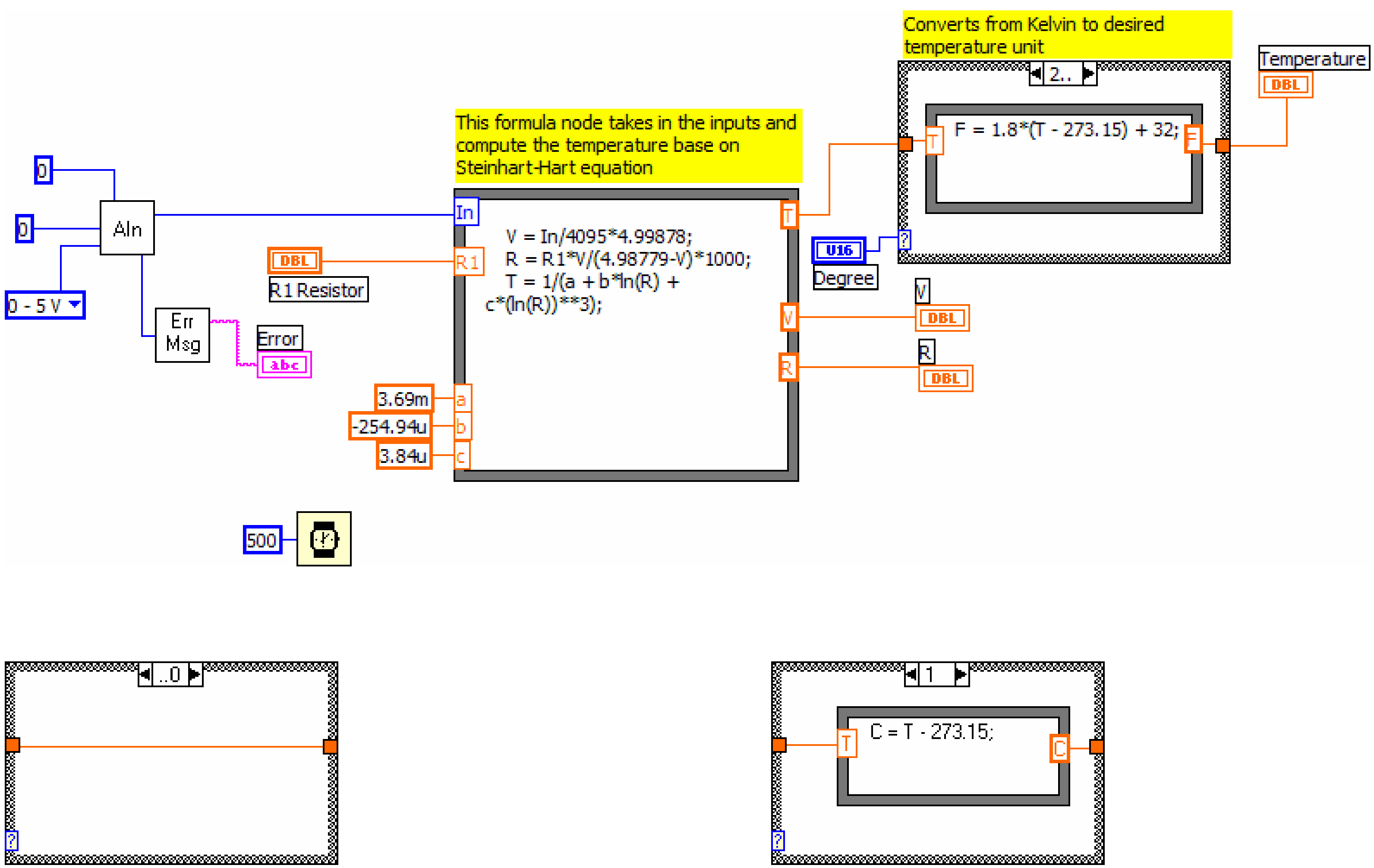
A.5.1 Thermistor 2 temp VI Connector Pane



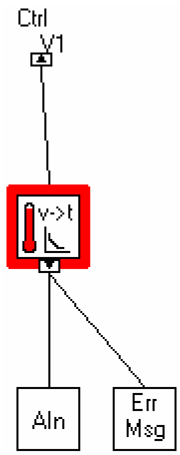

A.5.2 Thermistor 2 temp VI Front Panel



A.5.3 Thermistor 2 temp VI Block Diagram



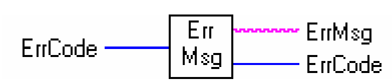
A.5.4 Thermistor 2 temp VI Position in Hierarchy



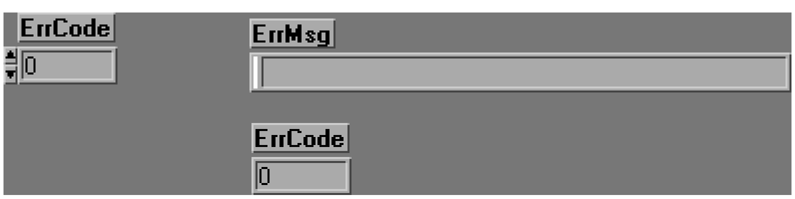

## A.6 ERRMSG.VI[3]

This is a third party VI from Measurement Computing. This VI checks the error code from the data acquisition card and converts it into an error message.

A.6.1 ERRMSG VI Connector Pane

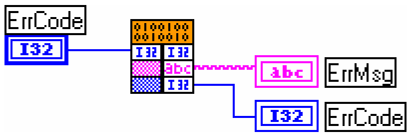



A.6.2 ERRMSG VI Front Panel

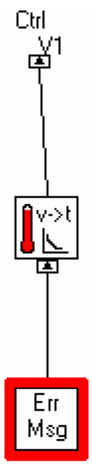



[3] Third party VI from Measurement Computing

A.6.3 ERRMSG VI Block Diagram



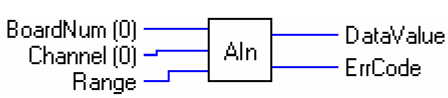

A.6.4 ERRMSG VI Position in Hierarchy




**A.7 AIn.VI[4]**

This is a third party VI from Measurement Computing. This VI obtains data from an analog channel and returns a 12-bit value. The 12-bit value is then used to compute the measured voltage.

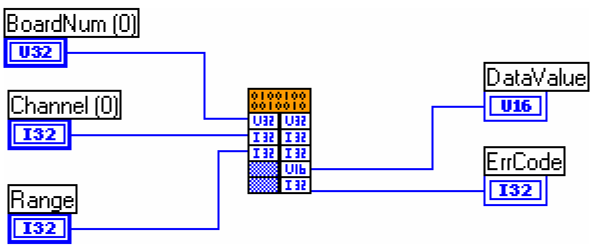
A.7.1 AIn VI Connector Pane



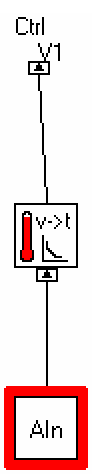

[4] Third party virtual instrument from Measurement Computing

A.7.2 AIn VI Front Panel


A.7.3 AIn VI Block Diagram



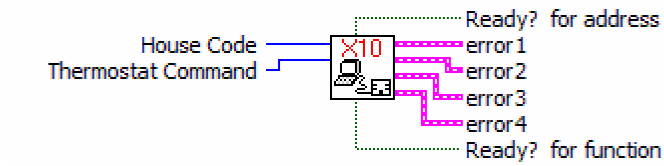

A.7.4 AIn VI Position in Hierarchy



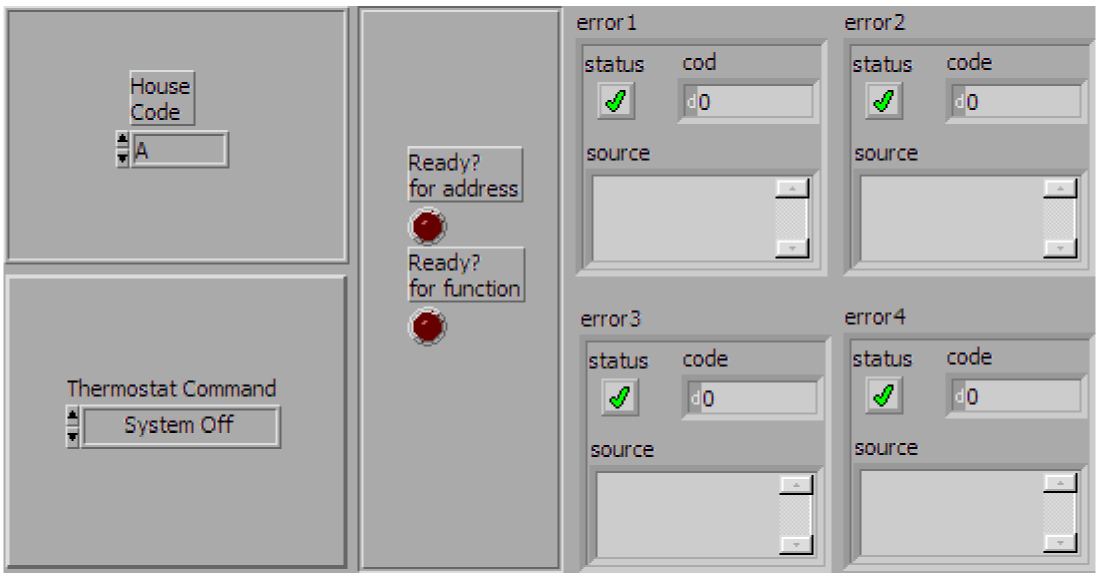

**A.8 x10-thermo.vi**

This VI converts and transmits X-10 thermostat commands.  This VI is very similar to X10v6 VI with changes to the X-10 commands sent.

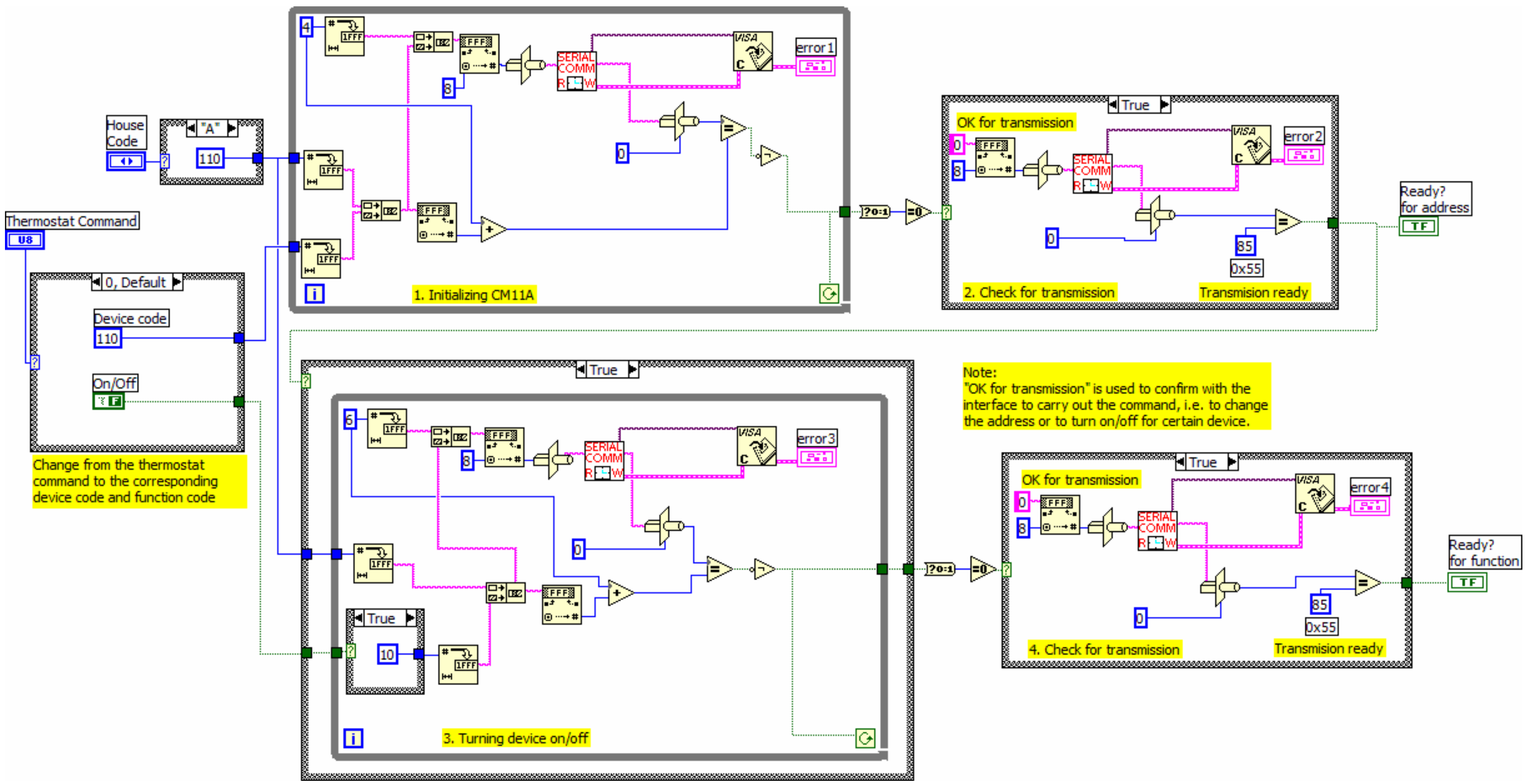
## A.8.1 x10-thermo VI Connector Pane



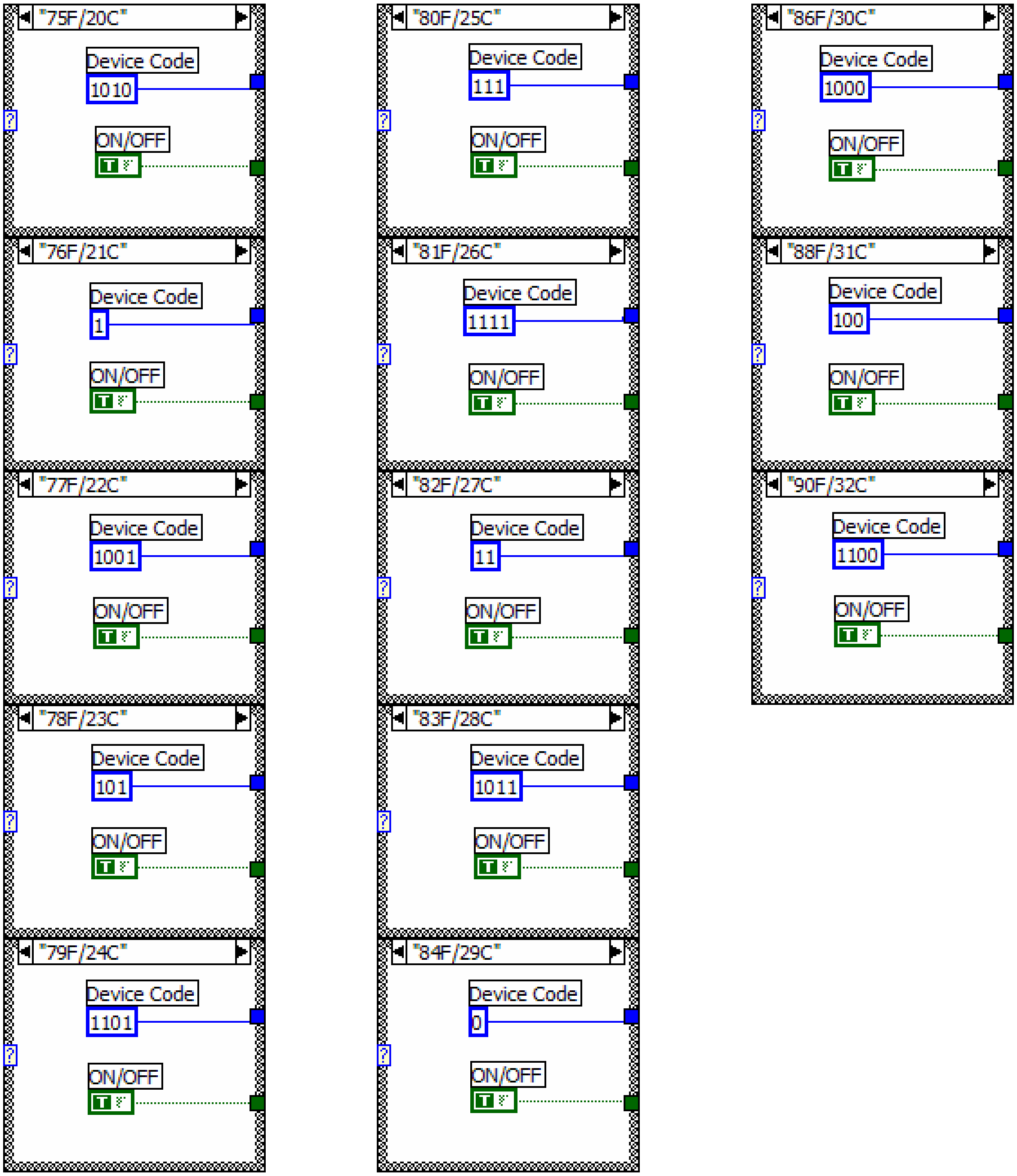

## A.8.2 x10-thermo VI Front Panel

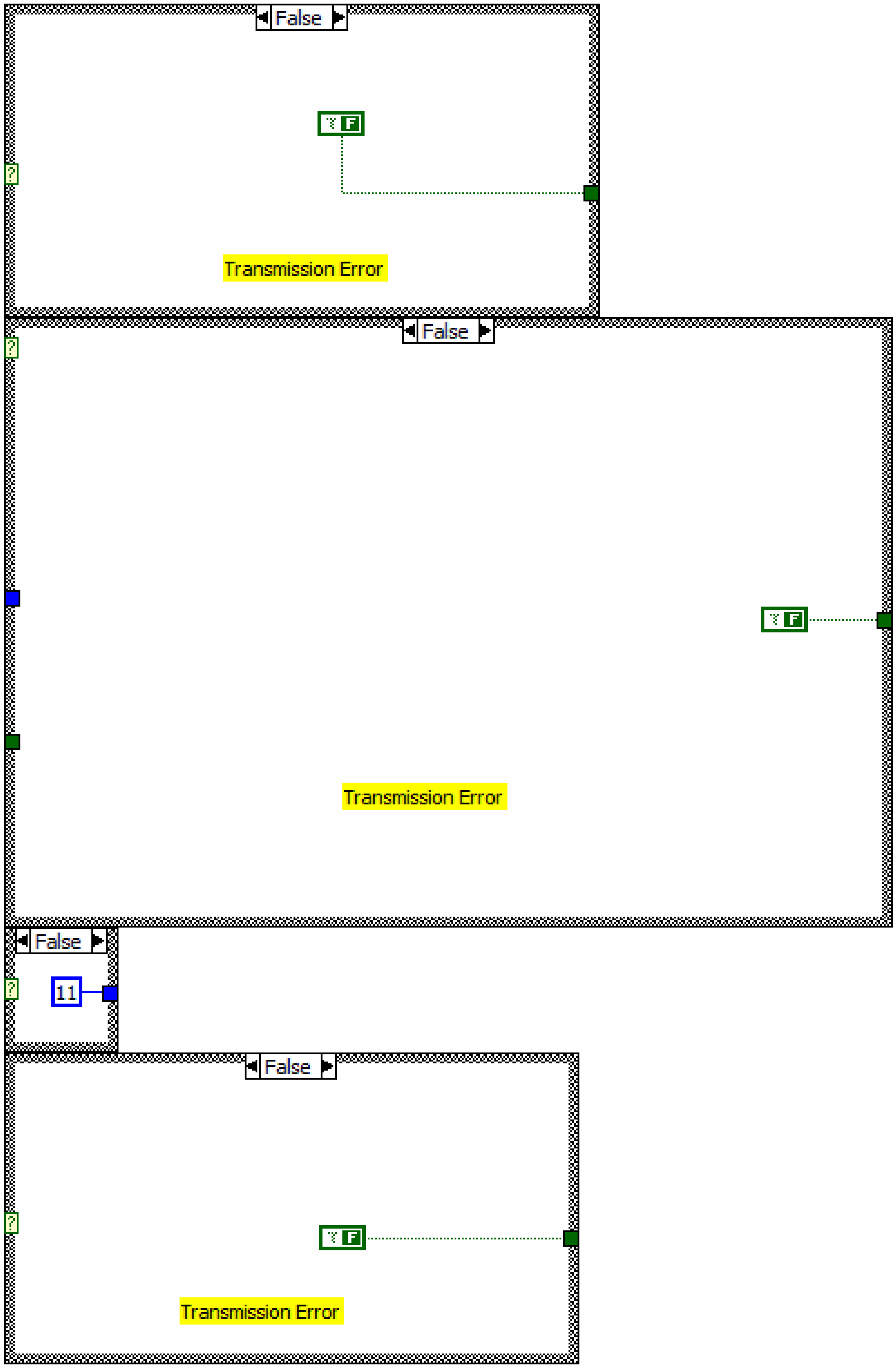


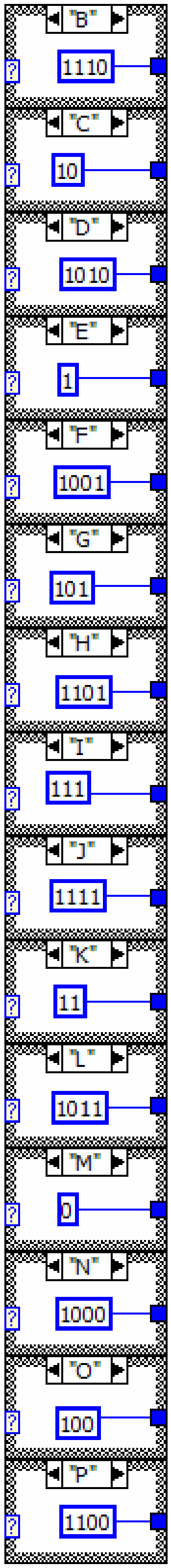
A.8.3 x10-thermo VI Block Diagram




"Heat Mode"
Device Code
1110
ON/OFF
"Cool Mode"
Device Code
10
ON/OFF
"Auto Mode"
Device Code
1010
ON/OFF
"40F/5C"
Device Code
1
ON/OFF
"60F/6C"
Device Code
1001
ON/OFF
"62F/7C"
Device Code
101
ON/OFF
"63F/8C"
Device Code
1101
ON/OFF
"64F/9C"
Device Code
111
ON/OFF
"65F/10C"
Device Code
1111
ON/OFF
"66F/11C"
Device Code
11
ON/OFF
"67F/12C"
Device Code
1011
ON/OFF
"68F/13C"
Device Code
0
ON/OFF
"69F/14C"
Device Code
1000
ON/OFF
"70F/15C"
Device Code
100
ON/OFF
"71F/16C"
Device Code
1100
ON/OFF
"72F/17C"
Device Code
110
ON/OFF
"73F/18C"
Device Code
1110
ON/OFF
"74F/19C"
Device Code
10
ON/OFF


"75F/20C"
Device Code
1010
ON/OFF
"76F/21C"
Device Code
1
ON/OFF
"77F/22C"
Device Code
1001
ON/OFF
"78F/23C"
Device Code
101
ON/OFF
"79F/24C"
Device Code
1101
ON/OFF
"80F/25C"
Device Code
111
ON/OFF
"81F/26C"
Device Code
1111
ON/OFF
"82F/27C"
Device Code
11
ON/OFF
"83F/28C"
Device Code
1011
ON/OFF
"84F/29C"
Device Code
0
ON/OFF
"86F/30C"
Device Code
1000
ON/OFF
"88F/31C"
Device Code
100
ON/OFF
"90F/32C"
Device Code
1100
ON/OFF


False
Transmission Error
False
Transmission Error
False
11
False
Transmission Error


"B"
1110
"C"
10
"D"
1010
"E"
1
"F"
1001
"G"
101
"H"
1101
"I"
111
"J"
1111
"K"
11
"L"
1011
"M"
0
"N"
1000
"O"
100
"P"
1100

# APPENDIX B

# VICTL AND CTLCGI2 VIRTUAL INSTRUMENT

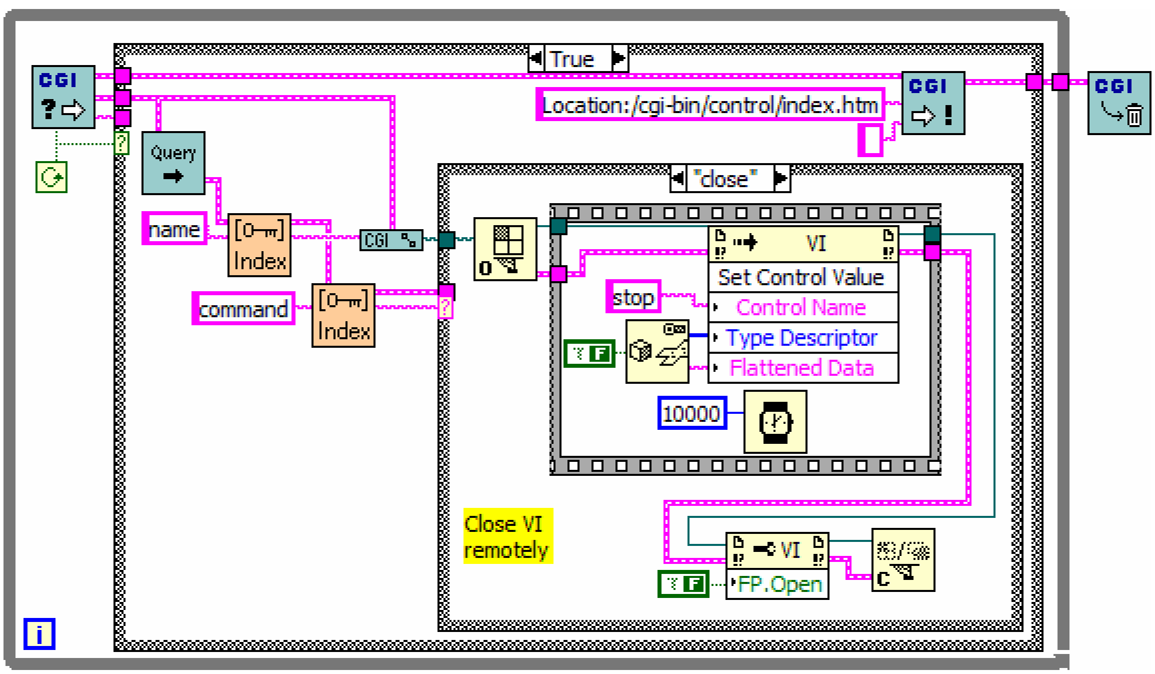
# SOURCE CODE

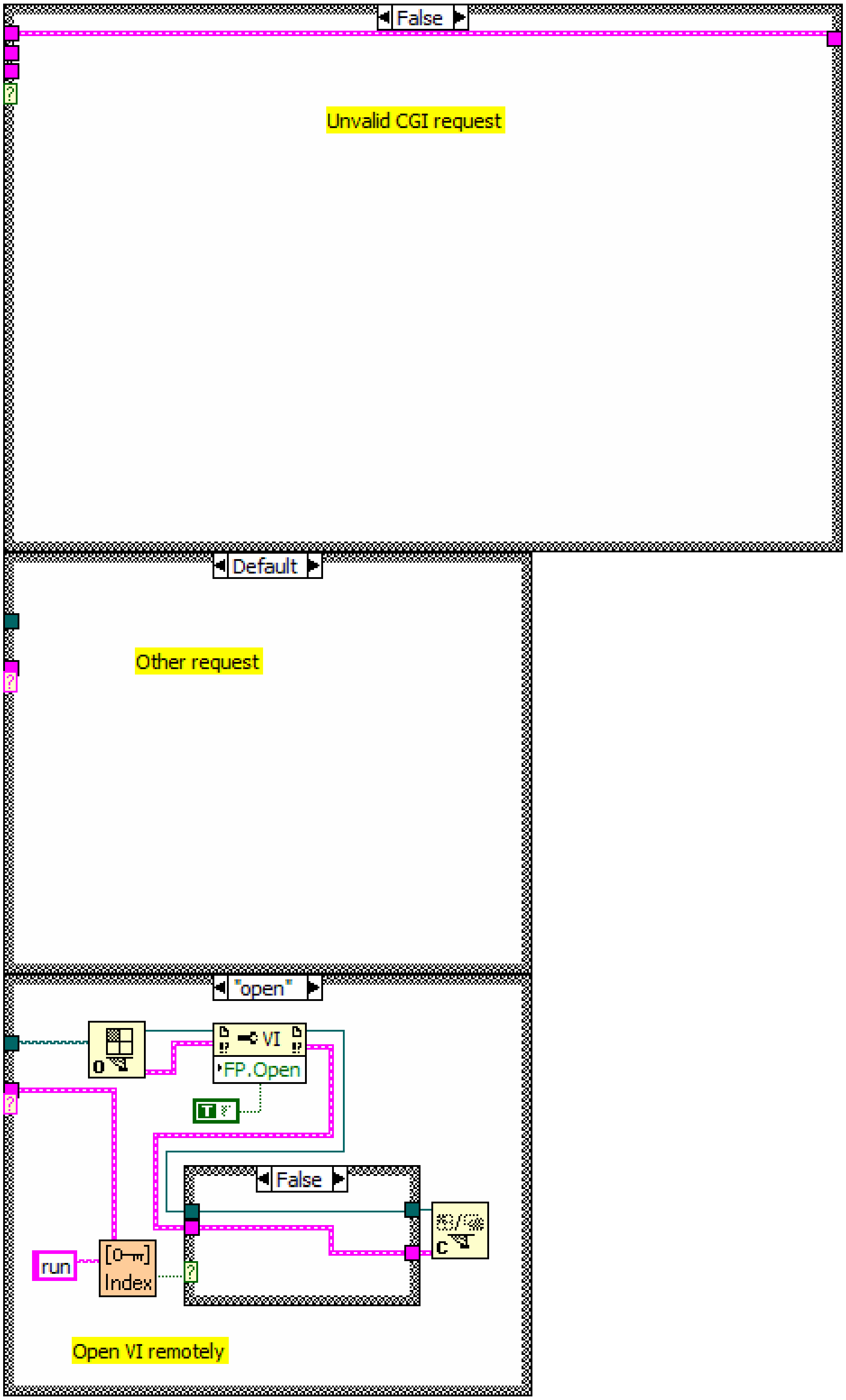
**B.1 victl.vi**

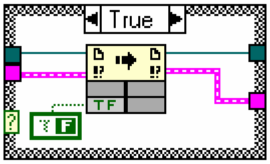
This VI is used to open or close other VI's remotely through CGI requests using a web browser.

B.1.1 victl VI Connector Pane


B.1.2 victl VI Block Diagram




False
Unvalid CGI request
Default
Other request
"open"
VI
FP.Open
False
run
Index
Open VI remotely

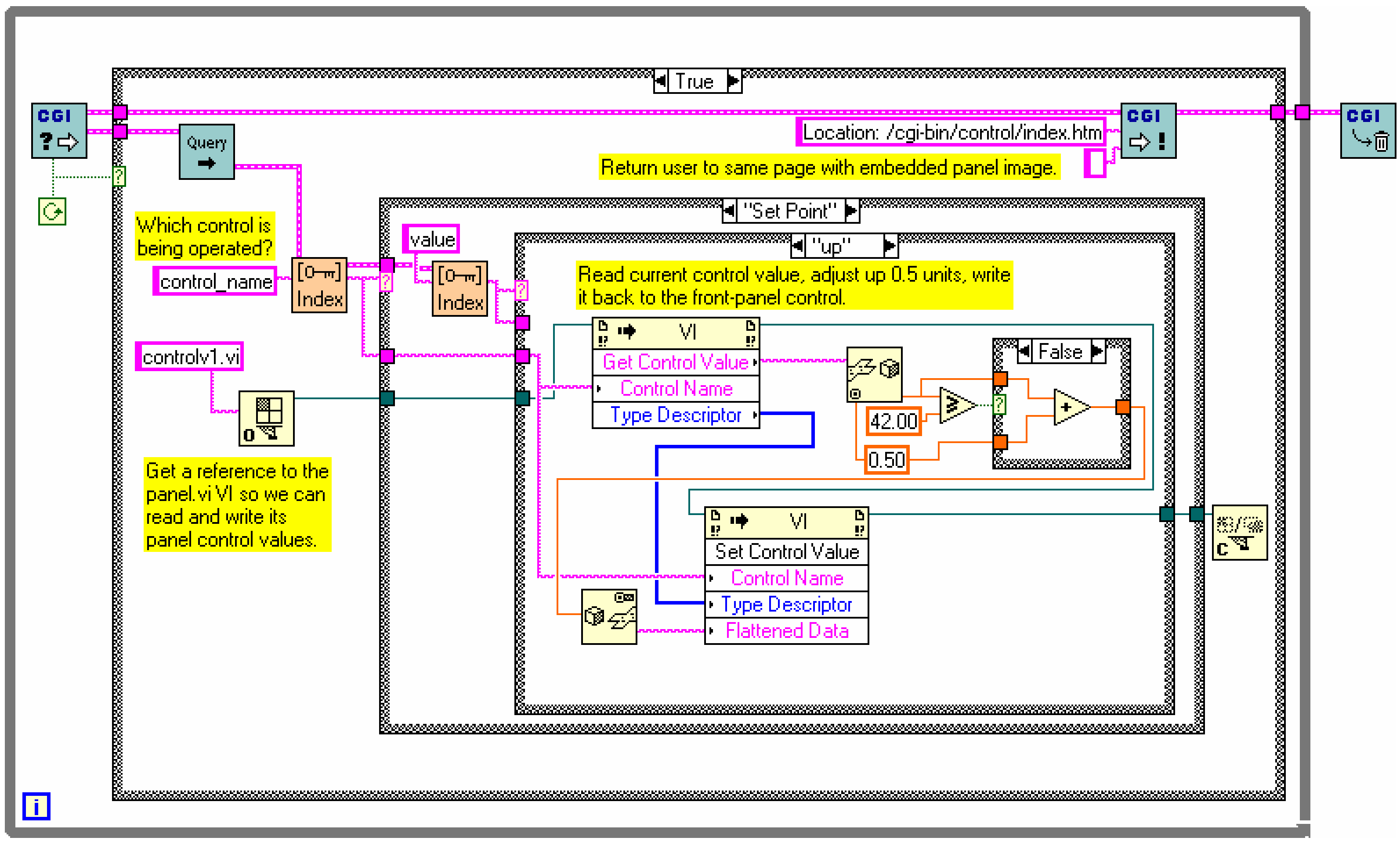


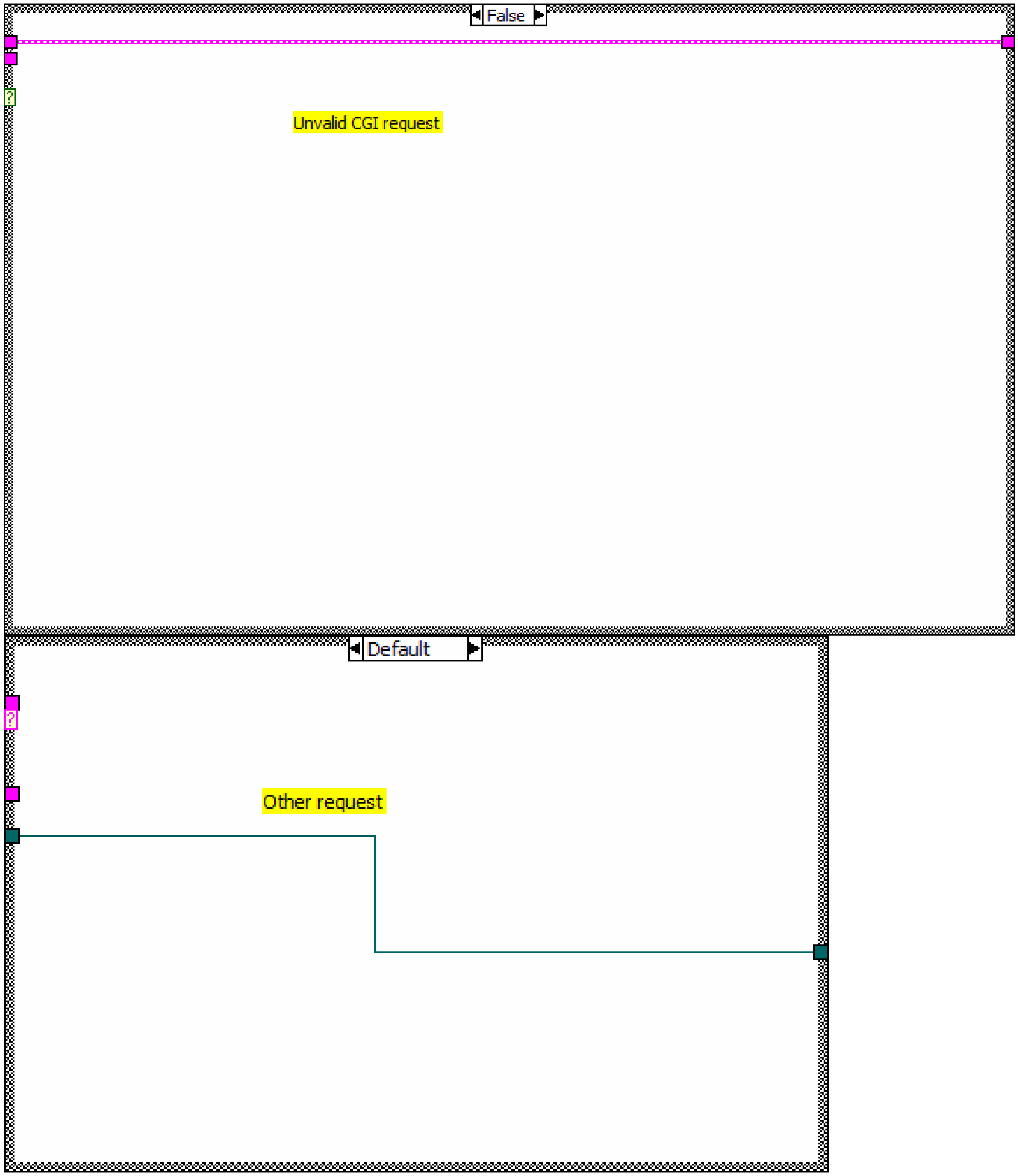

**B.2 ctlcgi2.vi**

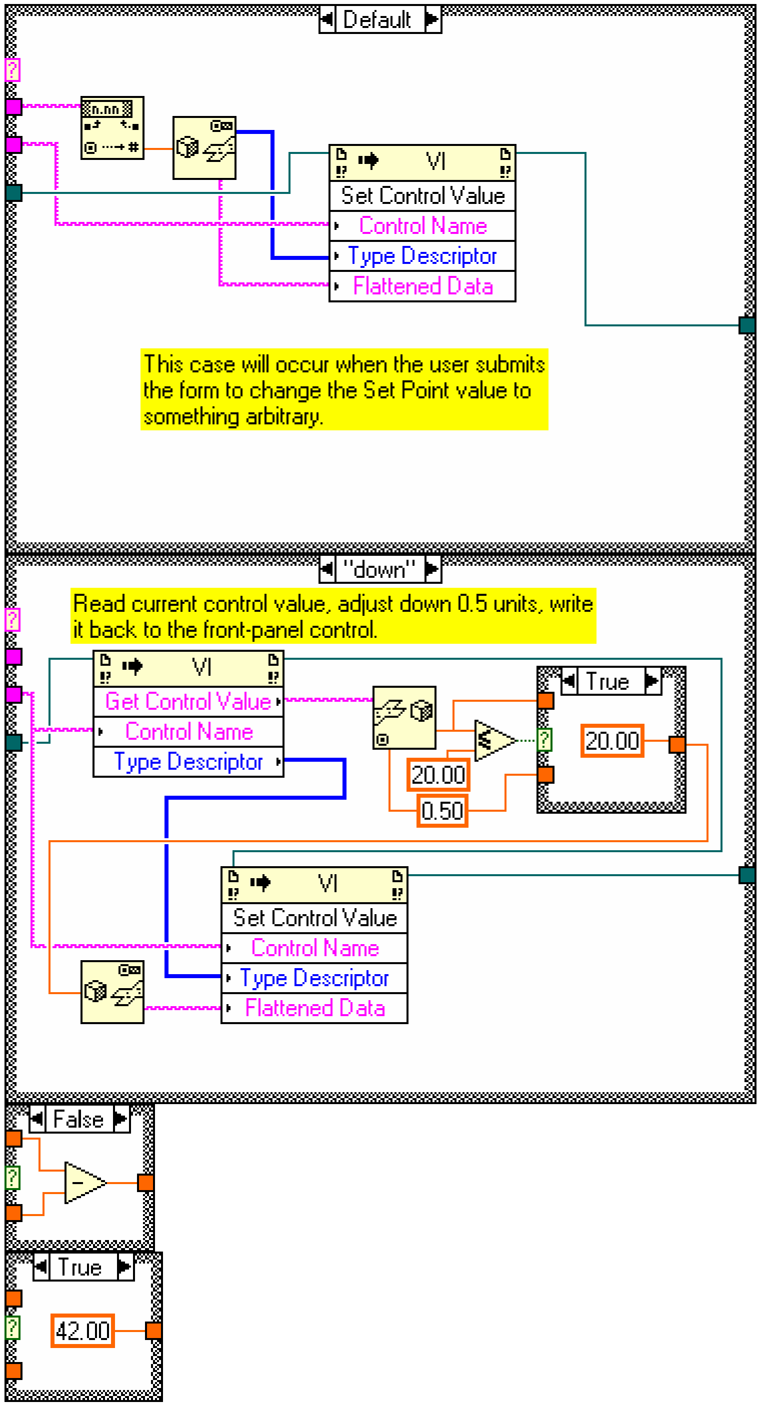
This VI is used to change the set point temperature for ControlV1 VI. With each up or down adjustment, the set point temperature changes by five degrees.

B.2.1 ctlcgi2 VI Connector Pane

B.2.2 ctlcgi2 VI Block Diagram




False
Unvalid CGI request
Default
Other request


Default
VI
Set Control Value
Control Name
Type Descriptor
Flattened Data
This case will occur when the user submits the form to change the Set Point value to something arbitrary.
"down"
Read current control value, adjust down 0.5 units, write it back to the front-panel control.
VI
Get Control Value
Control Name
Type Descriptor
True
20.00
20.00
0.50
VI
Set Control Value
Control Name
Type Descriptor
Flattened Data
False
True
42.00

# APPENDIX C

# OUTDOOR TEMPERATURE OBTAINED FROM NATIONAL WEATHER SERVICE TELECOMMUNICATIONS CENTER

| Time | | Date | | Temperature | | Dew Point | | Pressure | | Wind | | Weather |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EST | | | | (F) | (C) | (F) | (C) | Inches | (hPa) | MPH | | |
| 2 | PM | Jun | 18 | 91 | 32.78 | 62 | 16.67 | 30.13 | 1020 | ESE | 11 | |
| 3 | PM | Jun | 18 | 91 | 32.78 | 62 | 16.67 | 30.11 | 1019 | S | 3 | |
| 4 | PM | Jun | 18 | 91 | 32.78 | 62 | 16.67 | 30.1 | 1019 | S | 4 | |
| 5 | PM | Jun | 18 | 91 | 32.78 | 64 | 17.78 | 30.09 | 1018 | SSE | 4 | |
| 6 | PM | Jun | 18 | 89 | 31.67 | 66 | 18.89 | 30.09 | 1018 | SSE | 5 | |
| 7 | PM | Jun | 18 | 84 | 28.89 | 68 | 20.00 | 30.11 | 1019 | SSE | 3 | |
| 8 | PM | Jun | 18 | 78 | 25.56 | 69 | 20.56 | 30.12 | 1019 | Calm | | |
| 9 | PM | Jun | 18 | 77 | 25.00 | 69 | 20.56 | 30.13 | 1020 | SSE | 3 | |
| 10 | PM | Jun | 18 | 75 | 23.89 | 69 | 20.56 | 30.15 | 1020 | Calm | | |
| 11 | PM | Jun | 18 | 73 | 22.78 | 68 | 20.00 | 30.15 | 1020 | Calm | | |
| 12 | AM | Jun | 19 | 75 | 23.89 | 66 | 18.89 | 30.14 | 1020 | SE | 4 | |
| 1 | AM | Jun | 19 | 71 | 21.67 | 66 | 18.89 | 30.14 | 1020 | Calm | | |
| 2 | AM | Jun | 19 | 69 | 20.56 | 66 | 18.89 | 30.14 | 1020 | Calm | | |
| 3 | AM | Jun | 19 | 71 | 21.67 | 66 | 18.89 | 30.14 | 1020 | SE | 6 | |
| 4 | AM | Jun | 19 | 71 | 21.67 | 66 | 18.89 | 30.14 | 1020 | SE | 6 | |
| 5 | AM | Jun | 19 | 69 | 20.56 | 64 | 17.78 | 30.16 | 1021 | SE | 4 | |
| 6 | AM | Jun | 19 | 71 | 21.67 | 64 | 17.78 | 30.19 | 1022 | ESE | 4 | |
| 7 | AM | Jun | 19 | 75 | 23.89 | 66 | 18.89 | 30.22 | 1023 | ESE | 5 | |
| 8 | AM | Jun | 19 | 78 | 25.56 | 68 | 20.00 | 30.24 | 1024 | ESE | 6 | |
| 9 | AM | Jun | 19 | 80 | 26.67 | 69 | 20.56 | 30.24 | 1024 | SSE | 4 | |
| 10 | AM | Jun | 19 | 84 | 28.89 | 68 | 20.00 | 30.24 | 1024 | E | 8 | |
| 11 | AM | Jun | 19 | 86 | 30.00 | 66 | 18.89 | 30.23 | 1023 | E | 5 | |
| 12 | PM | Jun | 19 | 89 | 31.67 | 66 | 18.89 | 30.22 | 1023 | E | 12 | |
| 1 | PM | Jun | 19 | 89 | 31.67 | 62 | 16.67 | 30.2 | 1022 | E | 8 | |
| 2 | PM | Jun | 19 | 89 | 31.67 | 62 | 16.67 | 30.18 | 1022 | ENE | 10 | |
| 3 | PM | Jun | 19 | 89 | 31.67 | 60 | 15.56 | 30.16 | 1021 | ENE | 10 | |
| 4 | PM | Jun | 19 | 89 | 31.67 | 62 | 16.67 | 30.15 | 1020 | E | 5 | |
| 5 | PM | Jun | 19 | 87 | 30.56 | 62 | 16.67 | 30.15 | 1020 | E | 6 | |
| 6 | PM | Jun | 19 | 87 | 30.56 | 64 | 17.78 | 30.16 | 1021 | E | 3 | |
| 7 | PM | Jun | 19 | 82 | 27.78 | 64 | 17.78 | 30.16 | 1021 | Calm | | |
| 8 | PM | Jun | 19 | 77 | 25.00 | 64 | 17.78 | 30.15 | 1020 | ESE | 4 | |
| 9 | PM | Jun | 19 | 75 | 23.89 | 66 | 18.89 | 30.17 | 1021 | E | 3 | |
| 10 | PM | Jun | 19 | 75 | 23.89 | 66 | 18.89 | 30.19 | 1022 | SSE | 4 | |
| 11 | PM | Jun | 19 | 71 | 21.67 | 66 | 18.89 | 30.19 | 1022 | Calm | | |
| 12 | AM | Jun | 20 | 69 | 20.56 | 66 | 18.89 | 30.2 | 1022 | Calm | | |
| 1 | AM | Jun | 20 | 69 | 20.56 | 66 | 18.89 | 30.19 | 1022 | Calm | | |
| 2 | AM | Jun | 20 | 66 | 18.89 | 66 | 18.89 | 30.19 | 1022 | Calm | | |
| 3 | AM | Jun | 20 | 66 | 18.89 | 66 | 18.89 | 30.19 | 1022 | Calm | | |
| 4 | AM | Jun | 20 | 66 | 18.89 | 66 | 18.89 | 30.19 | 1022 | SE | 3 | |
| 5 | AM | Jun | 20 | 66 | 18.89 | 66 | 18.89 | 30.2 | 1022 | Calm | | |
| 6 | AM | Jun | 20 | 69 | 20.56 | 66 | 18.89 | 30.22 | 1023 | Calm | | |
| 7 | AM | Jun | 20 | 73 | 22.78 | 69 | 20.56 | 30.24 | 1024 | ESE | 3 | |
| 8 | AM | Jun | 20 | 77 | 25.00 | 69 | 20.56 | 30.25 | 1024 | SSE | 3 | |
| 9 | AM | Jun | 20 | 80 | 26.67 | 71 | 21.67 | 30.24 | 1024 | Calm | | |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | AM | Jun | 20 | 82 | 27.78 | 71 | 21.67 | 30.24 | 1024 | SSW | 3 | |
| 11 | AM | Jun | 20 | 82 | 27.78 | 71 | 21.67 | 30.23 | 1023 | Calm | | |
| 12 | PM | Jun | 20 | 84 | 28.89 | 69 | 20.56 | 30.21 | 1023 | E | 4 | |
| 1 | PM | Jun | 20 | 87 | 30.56 | 71 | 21.67 | 30.19 | 1022 | NNE | 5 | |
| 2 | PM | Jun | 20 | 87 | 30.56 | 69 | 20.56 | 30.17 | 1021 | Calm | | |
| 3 | PM | Jun | 20 | 87 | 30.56 | 69 | 20.56 | 30.15 | 1020 | Calm | | |
| 4 | PM | Jun | 20 | 89 | 31.67 | 68 | 20.00 | 30.14 | 1020 | Calm | | |
| 5 | PM | Jun | 20 | 84 | 28.89 | 68 | 20.00 | 30.14 | 1020 | S | 8 | |
| 6 | PM | Jun | 20 | 80 | 26.67 | 69 | 20.56 | 30.14 | 1020 | SW | 4 | |
| 7 | PM | Jun | 20 | 80 | 26.67 | 71 | 21.67 | 30.13 | 1020 | Calm | | |
| 8 | PM | Jun | 20 | 77 | 25.00 | 71 | 21.67 | 30.14 | 1020 | Calm | | |
| 9 | PM | Jun | 20 | 73 | 22.78 | 69 | 20.56 | 30.15 | 1020 | Calm | | |
| 10 | PM | Jun | 20 | 71 | 21.67 | 69 | 20.56 | 30.16 | 1021 | S | 4 | |
| 11 | PM | Jun | 20 | 71 | 21.67 | 69 | 20.56 | 30.15 | 1020 | S | 3 | |
| 12 | AM | Jun | 21 | 69 | 20.56 | 69 | 20.56 | 30.14 | 1020 | Calm | | |
| 1 | AM | Jun | 21 | 69 | 20.56 | 66 | 18.89 | 30.13 | 1020 | Calm | | |
| 2 | AM | Jun | 21 | 69 | 20.56 | 68 | 20.00 | 30.13 | 1020 | Calm | | |
| 3 | AM | Jun | 21 | 68 | 20.00 | 66 | 18.89 | 30.12 | 1019 | S | 3 | |
| 4 | AM | Jun | 21 | 68 | 20.00 | 66 | 18.89 | 30.14 | 1020 | Calm | | |
| 5 | AM | Jun | 21 | 68 | 20.00 | 66 | 18.89 | 30.14 | 1020 | S | 4 | |
| 6 | AM | Jun | 21 | 69 | 20.56 | 66 | 18.89 | 30.15 | 1020 | Calm | | |
| 7 | AM | Jun | 21 | 73 | 22.78 | 69 | 20.56 | 30.15 | 1020 | SW | 3 | |
| 8 | AM | Jun | 21 | 78 | 25.56 | 71 | 21.67 | 30.14 | 1020 | SSW | 3 | |
| 9 | AM | Jun | 21 | 80 | 26.67 | 71 | 21.67 | 30.15 | 1020 | SW | 5 | |
| 10 | AM | Jun | 21 | 82 | 27.78 | 71 | 21.67 | 30.14 | 1020 | SSW | 5 | |
| 11 | AM | Jun | 21 | 84 | 28.89 | 71 | 21.67 | 30.13 | 1020 | SSW | 5 | |
| 12 | PM | Jun | 21 | 87 | 30.56 | 71 | 21.67 | 30.1 | 1019 | SW | 6 | |
| 1 | PM | Jun | 21 | 89 | 31.67 | 71 | 21.67 | 30.06 | 1017 | Calm | | |
| 2 | PM | Jun | 21 | 89 | 31.67 | 71 | 21.67 | 30.04 | 1017 | WSW | 4 | |
| 3 | PM | Jun | 21 | 80 | 26.67 | 71 | 21.67 | 30.04 | 1017 | NW | 12 | thunder |
| 4 | PM | Jun | 21 | 77 | 25.00 | 71 | 21.67 | 30.03 | 1016 | SW | 3 | thunder in the vicinity |
| 5 | PM | Jun | 21 | 77 | 25.00 | 71 | 21.67 | 30.04 | 1017 | SE | 13 | |
| 6 | PM | Jun | 21 | 77 | 25.00 | 69 | 20.56 | 30 | 1015 | S | 5 | thunder in the vicinity |
| 7 | PM | Jun | 21 | 73 | 22.78 | 69 | 20.56 | 30 | 1015 | SSW | 4 | |
| 8 | PM | Jun | 21 | 71 | 21.67 | 69 | 20.56 | 30.01 | 1016 | Calm | | |
| 9 | PM | Jun | 21 | 71 | 21.67 | 69 | 20.56 | 30.03 | 1016 | S | 4 | |
| 10 | PM | Jun | 21 | 71 | 21.67 | 69 | 20.56 | 30.03 | 1016 | Calm | | |
| 11 | PM | Jun | 21 | 71 | 21.67 | 69 | 20.56 | 30.04 | 1017 | SSE | 3 | |
| 12 | AM | Jun | 22 | 71 | 21.67 | 69 | 20.56 | 30.02 | 1016 | Calm | | |
| 1 | AM | Jun | 22 | 71 | 21.67 | 69 | 20.56 | 30.04 | 1017 | WSW | 11 | thunder in the vicinity |
| 2 | AM | Jun | 22 | 71 | 21.67 | 68 | 20.00 | 30.01 | 1016 | NW | 8 | thunder in the vicinity |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 3 AM | Jun 22 | 69 | 20.56 | 68 | 20.00 | 30 | 1015 | NNW 6 | |
| 4 AM | Jun 22 | 69 | 20.56 | 68 | 20.00 | 29.98 | 1015 | N 5 | |
| 5 AM | Jun 22 | 69 | 20.56 | 68 | 20.00 | 29.99 | 1015 | NW 3 | |
| 6 AM | Jun 22 | 69 | 20.56 | 68 | 20.00 | 30.02 | 1016 | Calm | |

| Time | Date | Temperature | | Dew Point | | Pressure | | Wind | Weather |
|---|---|---|---|---|---|---|---|---|---|
| EST | | F | (C) | F | (C) | Inches | (hPa) | MPH | |
| 8 AM | Nov 26 | 44 | 6.67 | 39 | 3.89 | 30.21 | 1023 | NW 5 | |
| 9 AM | Nov 26 | 46 | 7.78 | 39 | 3.89 | 30.22 | 1023 | NNW 5 | |
| 10 AM | Nov 26 | 48 | 8.89 | 41 | 5.00 | 30.23 | 1023 | NW 3 | |
| 11 AM | Nov 26 | 46 | 7.78 | 42 | 5.56 | 30.24 | 1024 | WNW 3 | |
| 12 PM | Nov 26 | 46 | 7.78 | 42 | 5.56 | 30.23 | 1023 | NW 5 | |
| 1 PM | Nov 26 | 46 | 7.78 | 42 | 5.56 | 30.21 | 1023 | SW 3 | |
| 2 PM | Nov 26 | 46 | 7.78 | 42 | 5.56 | 30.22 | 1023 | W 5 | |
| 3 PM | Nov 26 | 46 | 7.78 | 42 | 5.56 | 30.21 | 1023 | NW 7 | |
| 4 PM | Nov 26 | 46 | 7.78 | 42 | 5.56 | 30.21 | 1023 | NNW 6 | |
| 5 PM | Nov 26 | 46 | 7.78 | 42 | 5.56 | 30.23 | 1023 | Calm | |
| 6 PM | Nov 26 | 46 | 7.78 | 42 | 5.56 | 30.23 | 1023 | Calm | |
| 7 PM | Nov 26 | 44 | 6.67 | 42 | 5.56 | 30.25 | 1024 | NNW 9 | |
| 8 PM | Nov 26 | 42 | 5.56 | 39 | 3.89 | 30.26 | 1024 | N 8 | |
| 9 PM | Nov 26 | 42 | 5.56 | 37 | 2.78 | 30.27 | 1025 | N 12 | |
| 10 PM | Nov 26 | 42 | 5.56 | 37 | 2.78 | 30.28 | 1025 | N 8 | |
| 11 PM | Nov 26 | 41 | 5.00 | 35 | 1.67 | 30.27 | 1025 | N 8 | |
| 12 AM | Nov 27 | 41 | 5.00 | 35 | 1.67 | 30.28 | 1025 | N 9 | |
| 1 AM | Nov 27 | 39 | 3.89 | 33 | 0.56 | 30.28 | 1025 | N 9 | |
| 2 AM | Nov 27 | 39 | 3.89 | 33 | 0.56 | 30.29 | 1025 | N 8 | |
| 3 AM | Nov 27 | 39 | 3.89 | 33 | 0.56 | 30.28 | 1025 | N 9 | |
| 4 AM | Nov 27 | 39 | 3.89 | 33 | 0.56 | 30.28 | 1025 | N 8 | |
| 5 AM | Nov 27 | 37 | 2.78 | 32 | 0.00 | 30.31 | 1026 | NW 5 | |
| 6 AM | Nov 27 | 35 | 1.67 | 32 | 0.00 | 30.32 | 1026 | N 6 | |
| 7 AM | Nov 27 | 35 | 1.67 | 30 | -1.11 | 30.33 | 1027 | N 7 | |
| 8 AM | Nov 27 | 35 | 1.67 | 28 | -2.22 | 30.35 | 1027 | N 7 | |
| 9 AM | Nov 27 | 37 | 2.78 | 28 | -2.22 | 30.37 | 1028 | N 8 | |
| 10 AM | Nov 27 | 42 | 5.56 | 28 | -2.22 | 30.39 | 1029 | NNW 9 | |
| 11 AM | Nov 27 | 44 | 6.67 | 28 | -2.22 | 30.39 | 1029 | N 14 | |
| 12 PM | Nov 27 | 44 | 6.67 | 26 | -3.33 | 30.37 | 1028 | N 14 | |
| 1 PM | Nov 27 | 46 | 7.78 | 26 | -3.33 | 30.35 | 1027 | NNW 12 | |
| 2 PM | Nov 27 | 46 | 7.78 | 26 | -3.33 | 30.32 | 1026 | NNW 12 | |
| 3 PM | Nov 27 | 46 | 7.78 | 26 | -3.33 | 30.31 | 1026 | N 9 | |
| 4 PM | Nov 27 | 44 | 6.67 | 26 | -3.33 | 30.32 | 1026 | NNW 9 | |
| 5 PM | Nov 27 | 41 | 5.00 | 26 | -3.33 | 30.33 | 1027 | NNW 7 | |
| 6 PM | Nov 27 | 39 | 3.89 | 26 | -3.33 | 30.32 | 1026 | NNW 3 | |
| 7 PM | Nov 27 | 37 | 2.78 | 26 | -3.33 | 30.33 | 1027 | NNW 5 | |
| 8 PM | Nov 27 | 35 | 1.67 | 24 | -4.44 | 30.35 | 1027 | NNW 5 | |
| 9 PM | Nov 27 | 33 | 0.56 | 26 | -3.33 | 30.36 | 1028 | NNE 7 | |
| 10 PM | Nov 27 | 32 | 0.00 | 26 | -3.33 | 30.35 | 1027 | N 5 | |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 11 | PM | Nov | 27 | 33 | 0.56 | 26 | -3.33 | 30.35 | 1027 | N | 5 | |
| 12 | AM | Nov | 28 | 32 | 0.00 | 26 | -3.33 | 30.35 | 1027 | N | 5 | |
| 1 | AM | Nov | 28 | 30 | -1.11 | 24 | -4.44 | 30.35 | 1027 | N | 5 | |
| 2 | AM | Nov | 28 | 28 | -2.22 | 24 | -4.44 | 30.36 | 1028 | Calm | | |
| 3 | AM | Nov | 28 | 26 | -3.33 | 24 | -4.44 | 30.36 | 1028 | Calm | | |
| 4 | AM | Nov | 28 | 26 | -3.33 | 24 | -4.44 | 30.37 | 1028 | NNW | 3 | |
| 5 | AM | Nov | 28 | 24 | -4.44 | 21 | -6.11 | 30.36 | 1028 | Calm | | |
| 6 | AM | Nov | 28 | 26 | -3.33 | 23 | -5.00 | 30.37 | 1028 | N | 5 | |
| 7 | AM | Nov | 28 | 26 | -3.33 | 24 | -4.44 | 30.41 | 1029 | W | 3 | |
| 8 | AM | Nov | 28 | 26 | -3.33 | 24 | -4.44 | 30.42 | 1030 | Calm | | |
| 9 | AM | Nov | 28 | 33 | 0.56 | 26 | -3.33 | 30.44 | 1030 | Calm | | |
| 10 | AM | Nov | 28 | 37 | 2.78 | 28 | -2.22 | 30.44 | 1030 | Calm | | |
| 11 | AM | Nov | 28 | 42 | 5.56 | 26 | -3.33 | 30.43 | 1030 | WNW | 5 | |
| 12 | PM | Nov | 28 | 46 | 7.78 | 26 | -3.33 | 30.4 | 1029 | NW | 6 | |
| 1 | PM | Nov | 28 | 48 | 8.89 | 26 | -3.33 | 30.36 | 1028 | WNW | 3 | |
| 2 | PM | Nov | 28 | 48 | 8.89 | 24 | -4.44 | 30.35 | 1027 | N | 8 | |
| 3 | PM | Nov | 28 | 50 | 10.00 | 26 | -3.33 | 30.33 | 1027 | NNW | 6 | |
| 4 | PM | Nov | 28 | 48 | 8.89 | 24 | -4.44 | 30.33 | 1027 | WNW | 5 | |
| 5 | PM | Nov | 28 | 44 | 6.67 | 24 | -4.44 | 30.32 | 1026 | NW | 3 | |
| 6 | PM | Nov | 28 | 39 | 3.89 | 26 | -3.33 | 30.32 | 1026 | Calm | | |
| 7 | PM | Nov | 28 | 33 | 0.56 | 28 | -2.22 | 30.33 | 1027 | Calm | | |
| 8 | PM | Nov | 28 | 33 | 0.56 | 26 | -3.33 | 30.33 | 1027 | Calm | | |
| 9 | PM | Nov | 28 | 30 | -1.11 | 26 | -3.33 | 30.33 | 1027 | Calm | | |
| 10 | PM | Nov | 28 | 30 | -1.11 | 24 | -4.44 | 30.31 | 1026 | SW | 3 | |
| 11 | PM | Nov | 28 | 30 | -1.11 | 26 | -3.33 | 30.33 | 1027 | Calm | | |
| 12 | AM | Nov | 29 | 30 | -1.11 | 24 | -4.44 | 30.31 | 1026 | Calm | | |
| 1 | AM | Nov | 29 | 30 | -1.11 | 24 | -4.44 | 30.28 | 1025 | S | 3 | |
| 2 | AM | Nov | 29 | 28 | -2.22 | 24 | -4.44 | 30.26 | 1024 | SSE | 6 | |
| 3 | AM | Nov | 29 | 26 | -3.33 | 24 | -4.44 | 30.25 | 1024 | Calm | | |
| 4 | AM | Nov | 29 | 30 | -1.11 | 24 | -4.44 | 30.26 | 1024 | SW | 3 | |
| 5 | AM | Nov | 29 | 30 | -1.11 | 24 | -4.44 | 30.26 | 1024 | SW | 3 | |
| 6 | AM | Nov | 29 | 30 | -1.11 | 24 | -4.44 | 30.23 | 1023 | S | 5 | |
| 7 | AM | Nov | 29 | 30 | -1.11 | 24 | -4.44 | 30.22 | 1023 | S | 5 | |
| 8 | AM | Nov | 29 | 33 | 0.56 | 26 | -3.33 | 30.22 | 1023 | SSW | 3 | |